DEDUCTION IN MATCHING LOGIC

ADAM FIEDLER

Master's Thesis
Faculty of Informatics
Masaryk University

May 2022

DECLARATION

Hereby I declare that this thesis is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during the elaboration of this work are properly cited and listed in complete reference to the due source.

Brno, May 2022

Adam Fiedler

ADVISOR: doc. Mgr. Jan Obdržálek, PhD.

CONSULTANT: Xiaohong Chen

"All good (constructive) logic must have an operational side."

— Jean-Yves Girard [15, p. 21]

ACKNOWLEDGMENTS

First, I would like to thank my advisor, doc. Mgr. Jan Obdržálek, PhD., for his continuous guidance and patience with me. My complicated writing style remains a problem. However, this thesis would be much more difficult to understand without his help. Always when I thought that something could not be explained more clearly, he proved me wrong. He gave me so many helpful writing suggestions that I have the feeling he read the thesis in more detail than I did.

Most warm-hearted thanks go to Xiaohong Chen for all the inspiring discussions we had throughout the year, his brilliant permutation idea, and his willingness to read all my drafts. Xiaohong taught me much about matching logic and research. His passion for science never fails to astonish me. I would also like to thank the rest of the FSL laboratory at the UIUC for taking me as their own and for their amazing work.

Last but not least, I would like to thank my family and Juliana for their understanding in these difficult couple of months. Writing is a great ordeal when other things have to be put aside. It is important to remember the support of those around the writer.

ABSTRACT

Matching logic (ML) is a logic designed for reasoning about programs by means of operational semantics. We investigate the foundations of matching logic and its proof systems suited for formal verification. We focus on System $\mathcal{H}$, which is complete w.r.t. most matching logic theories used in practice. A problem open for several years is whether System $\mathcal{H}$ is complete w.r.t. all theories. In this thesis, we identify a tractable if-and-only-if-condition for completeness of System $\mathcal{H}$ and exploit it to find new classes of complete theories. While solving the completeness problem, we review some existing results and answer related questions on expressiveness, consistency, and (un)satisfiability. For example, we show a detailed embedding of first-order logic in matching logic, prove the well-known compactness property for ML, and present a new technique of constructing canonical models for matching logic theories with equality. We also borrow some notions from first-order logic and study their properties in matching logic.

KEYWORDS

matching logic, first-order logic, proof systems, completeness, deduction, compactness, conservative extensions, consistency, satisfiability, canonical models

CONTENTS

1 INTRODUCTION

Matching logic (ML) [8, 11, 27] is a logic designed for reasoning about programs by means of operational semantics. We can define the operational semantics of a programming language as a *matching logic theory* and then derive operational behaviors in this logical theory [23]. The goal is to have a single source of truth in the form of operational semantics and use it *unchanged* to generate the entire toolkit (e.g., compiler, debugger, verifier, or state-space explorer) for the given programming language automatically [27, p. 3]. This is because we would like to verify programs with a minimal trust base that is consistent for all stages of the development process.

The current consensus is that operational semantics is too low-level to be used for practical formal verification [27, p. 3]. Many state-of-the-art formal methods thus rely on alternative semantics, various translations, or "ad hoc" techniques. Even if these methods are proved to be correct (and the proofs are themselves correct), each indirection creates a possibility where things can go wrong. ML was born in an effort to overcome the obstacles associated with operational semantics and use it directly: as a single point of reference. Operational semantics is usually easy to understand, scales well, and can be debugged and tested because it is executable [27, p. 3]. The $\mathbb{K}$ framework [28], based on matching logic, is proof that using operational semantics for both execution and verification is feasible. In a nutshell, $\mathbb{K}$ takes operational semantics of a language L as input and generates an interpreter and verifier for L as output. The list of programming languages successfully defined in $\mathbb{K}$ includes C [19], Java [6], and JavaScript [24]. The $\mathbb{K}$ verifier was able to verify several complex heap-manipulating programs in [29] at a performance level comparable with verifiers crafted for a specific language [27, p. 4].

ML has grown considerably since its introduction as a variant of first-order logic in [27]. It turned out to be well-suited not only for defining operational semantics but also for capturing *other logics*, which is another strong argument for using ML. ML can conveniently express and unify many popular logics as ML theories. The basic matching logic introduced in [27] is expressive enough to capture first-order logic and the well-known modal logic S5 [1, 3, 22]. In [11], matching logic was extended with the least fixpoint μ-binder. This allowed capturing many other logics such as first-order logic with least fixpoints, modal μ-logic [21], dynamic logic [17], as well as various temporal logics such as linear temporal logic [25] or reachability logic [26]. Another variant called applicative matching logic (AML) was introduced in [9]

to capture, e.g., the λ-calculus [9]. Matching logic can be seen as a logic unifying all of these different logics and as the driving force behind the development of $\mathbb{K}$.

Unlike axiomatic semantics that have to be tailored for each programming language on its own, ML has a single proof system called System $\mathcal{H}$ [11], which is language-independent. Given an operational semantics of a language, $\mathcal{H}$ can be used to derive the properties of programs written in this language. $\mathcal{H}$ is complete w.r.t. most matching logic theories used in practice, i.e., for most "useful" theories Γ we have

$$\Gamma \models \varphi \text{ implies } \Gamma \vdash_{\mathcal{H}} \varphi.$$

What these theories have in common is that they can define equality "=". The question of whether $\mathcal{H}$ is complete w.r.t. all theories, even for those without equality, has been open for several years [11, p. 1].

CONTRIBUTIONS. This thesis is a modest follow-up of [10, 11, 27] investigating the foundations of matching logic in order to answer the question of whether System $\mathcal{H}$ is complete. We take ML in its basic form (without the μ-binder) and ponder questions on the intersection of semantics and provability. The most notable of our contributions include:

(1) We present a full proof of embedding first-order logic in matching logic stronger than the original embedding presented in [27]. Namely, for every first-order S-theory Φ, we construct a matching logic theory Γ_S such that $\Phi \models_{\mathsf{FOL}} \varphi$ iff $\Phi \cup \Gamma_S \models_{\mathsf{ML}} \varphi$.

(2) We define a new concept called *equality extensions* and exploit it to identify a tractable if-and-only-if condition for completeness of $\mathcal{H}$.

(3) We reduce the problem of completeness to the problem of whether $\mathcal{H}$ is complete w.r.t. all *finite* theories.

(4) We prove that $\mathcal{H}$ is complete w.r.t. some particular classes of theories even without equality.

(5) We prove the well-known compactness theorem for ML.

(6) We borrow the notions of consistency and negation-complete theories from first-order logic and show that their ML counterparts behave similarly.

(7) We develop a new technique of constructing canonical models for theories with equality based on the work in [10].

In the process of dealing with the completeness problem, we also answer some related questions regarding expressiveness, (un)satisfiability, and consistency.

STRUCTURE OF THE THESIS. First, we get ourselves familiar with matching logic in Chapter 2, especially with concepts important for our aims. We discuss how matching logic relates to first-order logic in Chapter 3 and show that they have the same level of expressiveness. Chapter 4 reviews a proof system for matching logic called System $\mathcal{H}$ (and briefly also its predecessor System $\mathcal{P}$). In Chapter 5, we investigate completeness of $\mathcal{H}$, formulate an if-and-only-if condition for completeness, solve completeness w.r.t. some classes of theories even without equality, and solve related problems. Finally, the last Chapter 6 deals with canonical models in ML: we briefly overview existing results and show a new technique of constructing canonical models for theories with equality.

1.1 MATHEMATICAL CONVENTIONS

Let us declare a series of conventions that are followed throughout the thesis. Every block of mathematical text (definitions, examples, lemmas, theorems, etc.) ends with a black square "■", not just proofs; this way it is clear where a block ends even if it consists of multiple paragraphs. The symbol $\mathbb{N}$ stands for natural numbers, $\mathbb{Z}$ for integers. We always assume $0 \in \mathbb{N}$. When it is necessary to drop 0, we write $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$.

Lowercase letters from the beginning of the English alphabet such as a, b, c and m, n denote *elements* of a set (natural numbers, the carrier set of a model, ...). The letters i, j are reserved for indices we are currently considering in a list such as $a_1, \ldots, a_i, \ldots, a_n$ for some $n \in \mathbb{N}$. Note that the list $a_1, \ldots, a_n$ can be empty when $n = 0$. On the other hand, letters from the end of the English alphabet x, y, z or $x_1, \ldots, x_n$ are strictly used for *variables*. Lowercase Greek letters $\psi, \varphi, \xi, \gamma, \delta$ always denote formulas. Because we want to stay consistent with other matching logic publications, the letter ρ is reserved for valuations and σ is a placeholder formal symbol (similarly to how P is often used in first-order logic). If we use an "enclosing" formal symbol such as the ceiling symbol $\lceil \cdot \rceil$, by $\lceil \varphi \rceil$ we mean the application $\lceil \cdot \rceil(\varphi)$. We often use an underline to distinguish some formal symbols from their corresponding meta-object, e.g., the symbol $\underline{0}$ and the number $0 \in \mathbb{N}$.

If we use any stand-alone uppercase letter, we likely refer to a set of some kind. Uppercase Greek letters such as $\Psi, \Phi, \Xi, \Gamma, \Delta$ are specifically used for *sets of formulas*. Γ will stand for sets of matching logic formulas, and Φ for a set of FOL formulas. As usual, for a set A we write $P(A)$ to mean the *powerset* of A, and by

$$A^n = \underbrace{A \times \ldots \times A}_{n \text{ times}}$$

we mean the n-ary Cartesian power (note that $A^0 = \{\emptyset\}$). By $|A|$ we denote the cardinality of A. We assume all functions to be *total*, unless

stated otherwise. If $f : A \to B$, then $f|_{A'}$ for $A' \subseteq A$ is the restriction of f to A' defined as $f|_{A'} = f \cap (A' \times B)$. A *function composition* $g \circ f : A \to C$ of functions $f : A \to B$ and $g : B \to C$ is defined as $(g \circ f)(x) = g(f(x))$. We also differentiate between equality "$=$" and meta-level equality "$\equiv$". For example, $2 \cdot 21 = 42$ is different from $\exists x.\, x \equiv \exists y.\, y$, which means we consider α-equivalent formulas as the same. In ambiguous situations we also explicitly write $a := b$ instead of $a = b$ to highlight that we mean a is *defined* as b.

Finally, some note on special font styles. The calligraphic font styles $\mathcal{P}, \mathcal{H}$ or $\mathcal{D}$ are strictly used for *Hilbert-style proof systems*. The Fraktur $\mathfrak{A}, \mathfrak{B}, \mathfrak{M}$ is always used for *structures*, the letter $\mathfrak{I} = (\mathfrak{A}, v)$ for FOL interpretations (where v is an FOL valuation of variables). This is relatively standard in publications on logic [3, 4, 13, 14] and it helps to avoid defining redundant notation. We specifically follow the convention set by [13]: if $\mathfrak{A}$ is a structure, the corresponding letter A always denotes the *carrier set* of the structure and $a \in A$ its element.

1.2 A BRIEF REVIEW OF FIRST-ORDER LOGIC

Because we will often be discussing connections with first-order logic, let us informally recap basic definitions and our notation. First-order logic formulas are formal expressions over an alphabet consisting of a signature, logical connectives, quantifiers, and auxiliary symbols such as "(", ")", or ",". An FOL *signature* is a triple $S = (\textsc{Var}, \textsc{Pred}, \textsc{Func})$, where $\textsc{Pred} \cap \textsc{Func} = \emptyset$ and:

- $\textsc{Var}$ is a countably infinite set of variables.
- $\textsc{Pred} = \bigcup_{n \in \mathbb{N}^+} \textsc{Pred}_n$, where each $\textsc{Pred}_n$ is countable and we have $\textsc{Pred}_i \cap \textsc{Pred}_j = \emptyset$ for $i \neq j$.

  If $P \in \textsc{Pred}_n$, we call P an n-ary predicate symbol.
- $\textsc{Func} = \bigcup_{n \in \mathbb{N}} \textsc{Func}_n$, where each $\textsc{Func}_n$ is countable and we have $\textsc{Func}_i \cap \textsc{Func}_j = \emptyset$ for $i \neq j$.

  If $f \in \textsc{Func}_n$, we call f an n-ary function symbol.

Given a signature $S = (\textsc{Var}, \textsc{Pred}, \textsc{Func})$, we define a grammar of FOL S-terms t and S-formulas φ as follows:

$$t ::= x \in \textsc{Var} \mid f(t_1, \ldots, t_n) \text{ if } f \in \textsc{Func}_n$$
$$\varphi ::= P(t_1, \ldots, t_n) \text{ if } P \in \textsc{Pred}_n \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \exists x.\, \varphi \text{ if } x \in \textsc{Var}$$

Semantics of S-terms and S-formulas are given by interpretations $(\mathfrak{A}, v)$, where $v : \textsc{Var} \to A$ is a valuation of variables and $\mathfrak{A} = (A, (\cdot)^{\mathfrak{A}})$ is an S-structure, consisting of

- a non-empty carrier set A consisting of *elements*,
- a mapping $(\cdot)^{\mathfrak{A}}$ such that

- $f^{\mathfrak{A}} : M^n \to M$ for each function symbol $f \in \textsc{Func}_n$, and
- $P^{\mathfrak{A}} : M^n$ for each n-ary predicate symbol $P \in \textsc{Pred}_n$.

Given an interpretation $(\mathfrak{A}, v)$, an S-term t points to an element $v^{\mathfrak{A}}(t)$ and an S-formula φ is either true in $(\mathfrak{A}, v)$ ($(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi$) or false in $(\mathfrak{A}, v)$ ($(\mathfrak{A}, v) \not\models_{\mathsf{FOL}} \varphi$). We write $\mathfrak{A} \models_{\mathsf{FOL}} \varphi$ iff $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi$ for all valuations $v : \textsc{Var} \to A$. The definitions of $v^{\mathfrak{A}}(t)$ and $\models_{\mathsf{FOL}}$ are standard due to Tarski [30]. They can be found (with varying notation), e.g., in [13] or [14].

2 MATCHING LOGIC

Matching logic (ML)[1] is a variant of first-order logic (FOL) that does not differentiate between predicate symbols and function symbols [27]. Every FOL formula is an ML formula but not vice versa. For example, we can write an ML formula that says something about triples (n_1, n_2, n_3) such that $n_1 + n_2 = n_3$ as follows:

$$\varphi_{\text{SUM}} \equiv \exists x \exists y \exists z. \quad \underbrace{\langle x, \langle y, z \rangle \rangle}_{\text{structure}} \quad \wedge \quad \underbrace{x + y = z.}_{\text{logical constraint}}$$

Notice that φ_{SUM} is not an FOL formula; the reason is that $\langle \cdot, \cdot \rangle$ is neither a predicate symbol nor a function symbol. In ML, there is no distinction between predicates (formulas) and terms.

Formulas of ML are called *patterns*. A pattern is interpreted in ML as a set of model elements that "match" this pattern, similar to *pattern matching* in functional programming languages such as Haskell. To illustrate this, suppose we want to define a Haskell function `f` only for triples (x, y, z) of integers such that $x + y = z$. We could define `f` with guarded pattern matching as follows.

```
f :: (Integer, Integer, Integer) -> ...
f (x, y, z) | x + y == z = ...
```

The domain of the partial function `f` exactly matches the pattern φ_{SUM} defined in the first paragraph if we interpret symbols of φ_{SUM} in a reasonable manner. This means we can define an ML model $\mathfrak{N}$ of integers where φ_{SUM} is a pattern that $\mathfrak{N}$ interprets as the set

$$\{\underbrace{(n_1, n_2, n_3)}_{\text{structure}} \in \mathbb{Z}^3 \mid \underbrace{n_1 + n_2 = n_3}_{\text{logical constraint}}\} = \mathsf{dom}(\mathtt{f}).$$

Take $\mathfrak{N}$ as a black box for now. ML models are different from FOL models because formal symbols are interpreted in ML models as maps from elements to *sets* of model elements, not to model elements. On an abstract level, our model $\mathfrak{N}$ interprets $\langle \cdot, \cdot \rangle$ as a tuple constructor

$$\langle n_1, n_2 \rangle \mapsto_{\mathfrak{N}} \{n_1, \{n_1, n_2\}\}.$$

In this sense, $\langle x, \langle y, z \rangle \rangle$ matches triples $(n_1, n_2, n_3) \in \mathbb{Z}^3$, the pattern $x + y = z$ filters triples $(n_1, n_2, n_3) \in \mathbb{Z}^3$ such that $n_1 + n_2 = n_3$. Note that the pattern $x + y = z$ is interpreted as a set of elements as any other pattern. We discuss how equalities technically work in

1 We use the acronym ML throughout the thesis to mean matching logic, not the Meta Language due to Milner et al.

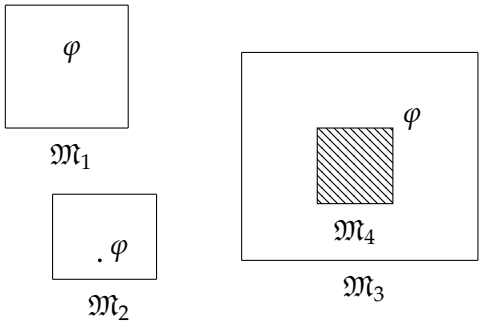


Figure 2.1: The pattern φ matches no elements in the model $\mathfrak{M}_1$, a single element in $\mathfrak{M}_2$, all elements of the set $M_4 \subset M_3$ in both $\mathfrak{M}_3$ and $\mathfrak{M}_4$. All elements of $\mathfrak{M}_4$ match φ, thus φ is valid in $\mathfrak{M}_4$.

Section 2.5. Finally, the connective $\wedge$ is interpreted as intersection $\cap$ of sets of elements matching $\langle x, \langle y, z\rangle\rangle$ and of those matching $x + y = z$.

Interpretations of ML symbols applied to a set of elements *distribute* over all contained elements. Let us give an example. If our model $\mathfrak{N}$ also interprets symbols even and prime as

$$\mathsf{even} \mapsto_{\mathfrak{N}} \{n \in \mathbb{N} \mid n \text{ is even}\},$$
$$\mathsf{prime} \mapsto_{\mathfrak{N}} \{p \in \mathbb{N} \mid p \text{ is prime}\},$$

then the pattern $\langle \mathsf{prime}, \langle \mathsf{prime}, \mathsf{even}\rangle\rangle$ distributes $\langle \cdot, \cdot \rangle$ over all elements matched by even and prime in the corresponding positions, i.e., it is interpreted by $\mathfrak{N}$ as

$$\{(p_1, p_2, n) \in \mathbb{Z}^3 \mid p_1, p_2 \text{ prime and } n \text{ even}\}.$$

Patterns can be meaningfully composed together exactly because they represent sets of model elements. This helps us avoid duplicities and redundant pattern definitions. For example, the pattern

$$\varphi_{\mathsf{GB}} \equiv \varphi_{\mathsf{SUM}} \wedge \langle \mathsf{prime}, \langle \mathsf{prime}, \mathsf{even}\rangle\rangle$$

matches some triple in $\mathfrak{N}$ for every even number greater than 2 iff *Goldbach's conjecture* is true. We did not need to change anything about the pattern φ_{SUM} from the first paragraph. ML semantics make structural reasoning very compact and composable (modular). That is why ML is well-suited for operational semantics of programming languages and formal verification using these semantics, which was one of the motivations behind introducing ML [27]. Readers interested in how to use ML for formal verification are referred to [29] for details.

The pattern φ_{GB} defined above also illustrates the dual character of patterns; patterns can specify sets of model elements as well as specify models among other models (Figure 2.1). This is different from FOL formulas. Consider some closed FOL formulas ψ_1, ψ_2; then the notation

$$\{\psi_1\} \not\models_{\mathsf{FOL}} \psi_2,$$

says that there is an FOL model of the FOL theory $\{\psi_1\}$ where ψ_2 does not hold. Here ψ_1 "specifies" FOL models we are considering, ψ_2 is an untrue statement about those models. FOL formulas express properties of models by referring to model elements with terms, where properties are either true or false. On the other hand, the notation

$$\{\psi_1\} \not\models_{\mathsf{ML}} \psi_2,$$

says that there is an ML model of $\{\psi_1\}$ where ψ_2 does not match *all elements* of the model. As we shall see in Section 2.2, if a pattern matches all elements of a model, the pattern is called *valid* in the model. This is how we use patterns as both terms referring to sets of model elements and formulas that are either valid (matching all elements) or not valid (not matching all elements).

Patterns have a dual character.

2.1 SYNTAX

Formulas of matching logic are called patterns. Analogously to other logics, patterns (formulas) are formal expressions that contain variables and formal symbols given by a *signature*.

Definition 2.1.1 (Signature). A *matching logic signature* (or simply signature) is a pair $(\textsc{Var}, \Sigma)$, where

- $\textsc{Var}$ is a countably infinite set of variables,
- $\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$ is a set of pairwise disjoint sets of symbols, where each Σ_n contains countably many symbols.

If $\sigma \in \Sigma_n$, then σ is called an n-ary symbol. ■

Notice that a signature does not differentiate between predicate or function symbols. A signature only declares what variables and formal symbols we use and of what arity each formal symbol is. Patterns (ML formulas) are then built with these variables and formal symbols applied to other patterns. We can also combine patterns by standard logical connectives.

Definition 2.1.2 (Pattern). Let $(\textsc{Var}, \Sigma)$ be an ML signature. Then a $(\textsc{Var}, \Sigma)$-pattern (or simply pattern) φ is any expression generated by the grammar

$$\begin{aligned}\varphi ::= {} & x \in \textsc{Var} \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \exists x.\, \varphi \text{ if } x \in \textsc{Var} \\ & \mid \sigma(\varphi_1, \ldots, \varphi_n) \text{ if } \sigma \in \Sigma_n.\end{aligned}$$

The set of all $(\textsc{Var}, \Sigma)$-patterns is denoted $\textsc{Pattern}_{(\textsc{Var},\Sigma)}$. We say that a pattern φ is in the signature $(\textsc{Var}, \Sigma)$ iff $\varphi \in \textsc{Pattern}_{(\textsc{Var},\Sigma)}$. ■

Note that $x \in \textsc{Var}$ is both a variable and a pattern. Unless stated otherwise, we assume $\textsc{Var}$ to be letters from the end of the English

alphabet such as x, y, z. That is why we usually drop this conventional VAR in the signature $(\textsc{Var}, \Sigma)$. The set of all Σ-patterns with conventional variables is denoted simply $\textsc{Pattern}_\Sigma$.

See Section 1.1 for other conventions.

By $\sigma \in \Sigma$ we slightly abuse notation to mean that $\sigma \in \Sigma_i$ for some $i \in \mathbb{N}$. We also write, e.g., $\Sigma = \{\lambda, \sigma(\cdot), \sigma'(\cdot,\cdot)\}$ to mean $\Sigma = \{\Sigma_0, \Sigma_1, \Sigma_2, \ldots\}$ where $\Sigma_0 = \{\lambda\}$, $\Sigma_1 = \{\sigma\}$, $\Sigma_2 = \{\sigma'\}$ and $\Sigma_n = \emptyset$ for $n > 2$. Whenever $\lambda \in \Sigma_0$, we call λ a *constant* (symbol). In patterns we write constants as λ instead of $\lambda()$.

Even though precedence parentheses "(" and ")" are not part of ML syntax, we use them to explicitly show pattern structure. For example, we need to distinguish $\neg\varphi \wedge \psi$ from $\neg(\varphi \wedge \psi)$. Negation $\neg$ always binds more tightly than other connectives, i.e.,

$$\neg\varphi \wedge \psi \equiv (\neg\varphi) \wedge \psi.$$

If Γ_{fin} is a finite set of patterns, we also make our lives slightly easier and write $\bigwedge \Gamma$ to mean the pattern $\bigwedge_{\gamma \in \Gamma_{\mathsf{fin}}} \gamma$.

BOUND VARIABLES. The scope of "$\exists$" goes as far as possible to the right unless it is limited by parentheses. For example,

$$\exists x.\, \psi \to ((\exists y.\, \varphi) \to z) \equiv \exists x.\, (\psi \to ((\exists y.\, \varphi) \to z)).$$

Analogously to FOL, "$\exists$" is a binder. Therefore *bound variables*, *free variables*, *capture-avoiding substitution* and *α-renaming* are defined accordingly (see, e.g., [27, p. 7]). By $\mathsf{FV}(\varphi)$ we denote the set of free variables of a pattern φ. When $\mathsf{FV}(\varphi) = \emptyset$, we say that φ is *closed*. We consider α-equivalent patterns to be *the same*, i.e., $\varphi \equiv \varphi'$ if φ, φ' are α-equivalent. Given a pattern $\varphi \in \textsc{Pattern}_\Sigma$, then

$$\varphi(x_1, \ldots, x_n) \text{ means } \mathsf{FV}(\varphi) \subseteq \{x_1, \ldots, x_n\}.$$

We write $\varphi[\psi/x]$ to mean capture-avoiding substitution[2], i.e., the result of substituting ψ for every *free* occurrence of x in φ with implicit α-renaming that prevents variable capture. As usual, the notation $\varphi[\psi_1/x_1, \psi_2/x_2, \ldots, \psi_n/x_n]$ for distinct x_i $(1 \leq i \leq n)$ means *simultaneous* capture-avoiding substitution.

REMARK ON SORTS. Unlike the canonical paper on ML [27], we only use single-sorted signatures. When we use a result from many-sorted ML, this is justified because we can simply assume that we only have one sort s. In fact, the single-sorted variant of ML we use in this thesis is as expressive as the original many-sorted ML. An example of how to define sorts in ML can be found in [8]. This is why sorts are often dropped in recent matching logic publications.

2 Note that ψ in $\varphi[\psi/x]$ can be any pattern as there is technically no difference between predicates and terms in matching logic.

2.2 SEMANTICS

Matching logic semantics is similar to *pattern matching* from functional programming languages such as Haskell. Intuitively speaking, a pattern is interpreted in ML models as a set of those model elements that *match* this pattern (hence the name matching logic) [11, 27]. For example, $(2,3)$ matches the pattern $\langle x, \underline{3} \rangle$ in a model of $\mathbb{N}^2$ if we interpret $x := 2$. If we do not care about the value of x, we can consider the pattern $\exists x.\, \langle x, \underline{3} \rangle$ that matches all elements of the set $\{(n,3) \mid n \in \mathbb{N}\}$ in a model of $\mathbb{N}^2$.

Let us give an intuition of semantics for each case. The pattern x (for any $x \in \textsc{Var}$) matches exactly one element given by a valuation (Definition 2.2.2); $\varphi_1 \wedge \varphi_2$ matches elements matching both φ_1 and φ_2; $\neg\varphi$ matches elements not matching φ; $\exists x.\, \varphi$ matches every element matching φ for some valuation of x. Finally, the pattern $\sigma(\varphi_1, \ldots, \varphi_n)$ matches elements determined by the *interpretation* of the symbol σ given by a model.

Definition 2.2.1 (Model). Let Σ be a signature. A matching logic Σ-model (or simply model) is a pair $\mathfrak{M} = (M, \{\sigma^{\mathfrak{M}}\}_{\sigma \in \Sigma})$ consisting of

- a non-empty carrier set M (often called the *domain*),
- a function $\sigma^{\mathfrak{M}} : M^n \to P(M)$ for every n-ary symbol $\sigma \in \Sigma_n$ (called *interpretation* of σ).

Every $m \in M$ is called an element of $\mathfrak{M}$ (or simply model element). ■

Even though technically $M^0 = \{\emptyset\}$, we interpret $\lambda^{\mathfrak{M}} : M^0 \to P(M)$ simply as $\lambda^{\mathfrak{M}} : P(M)$. We often overload $\sigma^{\mathfrak{M}}$ and mean the extension of $\sigma^{\mathfrak{M}}$ to *sets*, i.e., $\sigma^{\mathfrak{M}} : P(M)^n \to P(M)$ where

$$\sigma^{\mathfrak{M}}(A_1, \ldots, A_n) = \bigcup_{a_1 \in A_1, \ldots, a_n \in A_n} \sigma^{\mathfrak{M}}(a_1, \ldots, a_n).$$

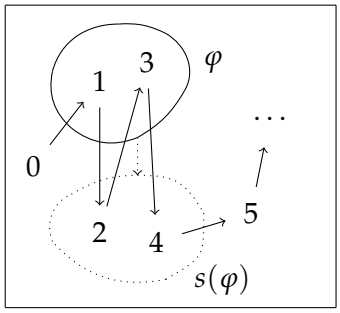
We also use notation in the following manner

$$\mathfrak{M} : M = \{m_1, m_2, \ldots\}, \sigma^{\mathfrak{M}}(m) = A$$

for $m \in M, A \subseteq M$ to mean $\mathfrak{M} = (\{m_1, m_2, \ldots\}, \{\ldots, \sigma^{\mathfrak{M}}, \ldots\})$ where $\sigma^{\mathfrak{M}}(m) = A$ to avoid unnecessary details.

Pattern matching in matching logic is formalized using the notion of valuation. For a given pattern, a valuation returns a set of model elements that match the given pattern.

Definition 2.2.2 (Valuation). Let $\mathfrak{M}$ be a Σ-model. A function $\rho : \textsc{Var} \to M$ is called an M-valuation. Given an M-valuation ρ, we define *pattern valuation* $\rho^{\mathfrak{M}} : \textsc{Pattern}_\Sigma \to P(M)$ for all $x \in \textsc{Var}$, all $\varphi \in \textsc{Pattern}_\Sigma$ and all $\sigma \in \Sigma$ inductively as follows:



$\mathfrak{M} = (\mathbb{N}, \{s_M\})$

Figure 2.2: Matching the pattern $\varphi \equiv (\underline{1} \vee \underline{2} \vee \underline{3}) \wedge \neg x)$ with $\rho(x) = 2$ in a model of natural numbers (Example 2.2.1). Arrows depict $s^{\mathfrak{M}}$; the dotted arrow depicts $\rho^{\mathfrak{M}}(s(\varphi)) = s^{\mathfrak{M}}(\{1,3\}) = \{2,4\}$.

- $\rho^{\mathfrak{M}}(x) = \{\rho(x)\}$,
- $\rho^{\mathfrak{M}}(\neg\varphi) = M \setminus \rho^{\mathfrak{M}}(\varphi)$,
- $\rho^{\mathfrak{M}}(\varphi_1 \wedge \varphi_2) = \rho^{\mathfrak{M}}(\varphi_1) \cap \rho^{\mathfrak{M}}(\varphi_2)$,
- $\rho^{\mathfrak{M}}(\exists x.\, \varphi) = \bigcup_{m \in M} \rho[m/x]^{\mathfrak{M}}(\varphi)$,
- $\rho^{\mathfrak{M}}(\sigma(\varphi_1, \ldots, \varphi_n)) = \sigma^{\mathfrak{M}}(\rho^{\mathfrak{M}}(\varphi_1), \ldots, \rho^{\mathfrak{M}}(\varphi_n))$ if $\sigma \in \Sigma_n$,

where $\rho[m/x](x) = m$ and $\rho[m/x](y) = \rho(y)$ for all $y \neq x$. We say that φ evaluates to A (with ρ) if $\rho^{\mathfrak{M}}(\varphi) = A$. We say that *a matches* φ or φ *matches* a (with $x := m$) if $a \in \rho[x/m]^{\mathfrak{M}}(\varphi)$ for some ρ. ■

Notice that M-valuations are maps from variables to elements of $\mathfrak{M}$. Similarly to other logics, we then extend M-valuations to valuations of patterns (formulas), which also depend on interpretations of symbols in the model $\mathfrak{M}$. Logical connectives correspond to basic operations over sets, symbols are arbitrary maps given by models from elements to sets of elements. This is illustrated by the following example.

Example 2.2.1 (Natural numbers). Consider the $\{\underline{0}, s(\cdot)\}$-model (depicted in Figure 2.2) defined as

$$\mathfrak{M} : M = \mathbb{N}, \underline{0}^{\mathfrak{M}} = \{0\}, s^{\mathfrak{M}}(n) = \{n+1\}.$$

Let us also define the expected syntactic sugar $\underline{n} \equiv \overbrace{s(s(\ldots s}^{n \text{ times}}(\underline{0})))$ for all $n \in \mathbb{N}^+$ and $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$.

It is easy to see for every M-valuation ρ that the pattern $\underline{n}$ matches the corresponding natural number n, i.e., $\rho^{\mathfrak{M}}(\underline{n}) = \{n\}$. Disjunctions evaluate to unions of natural numbers, conjunctions evaluate to intersections of matched numbers. For example, if $\rho(x) = 2$ then

$$\rho^{\mathfrak{M}}((\underline{1} \vee \underline{2} \vee \underline{3}) \wedge \neg x) = \{1,3\}$$
$$\Big((\underline{1} \vee \underline{2} \vee \underline{3}) \wedge \neg x \text{ matches 1 and 3 for } x := 2\Big).$$

The interpretation $s^{\mathfrak{M}}(X)$ applied to a subset of natural numbers $X \subseteq \mathbb{N}$ distributes $s^{\mathfrak{M}}$ over all elements of X, e.g.,

$$s^{\mathfrak{M}}(\{n_1, \ldots, n_k\}) = \bigcup_{1 \leq i \leq k} s^{\mathfrak{M}}(n_i) = \{n_1 + 1, \ldots, n_k + 1\}.$$

■

Of course, M-valuations have no effect on valuations of closed patterns, i.e., closed patterns match the same elements no matter how we interpret variables. We use this argument several times in the thesis, so it is useful to have it stated properly. This intuition is a corollary of the following proposition.[3]

Proposition 2.2.1 ([27]). Let $\mathfrak{M}$ be a Σ-model and $\varphi \in \text{PATTERN}_\Sigma$. For every M-valuation ρ_1, ρ_2 we have $\rho_1|_{\mathsf{FV}(\varphi_1)} = \rho_2|_{\mathsf{FV}(\varphi_2)}$ implies $\rho_1^{\mathfrak{M}}(\varphi_1) = \rho_2^{\mathfrak{M}}(\varphi_2)$. ■

As a concluding remark, we sometimes need to work with models restricted to smaller signatures. These are important when proving results related to patterns that contain only some subset of symbols in a given signature.

Definition 2.2.3 (Model restriction). Let $\mathfrak{M} = (M, I)$ be a Σ'-model and Σ be some signature such that $\Sigma \subseteq \Sigma'$. We define the *model restriction* of $\mathfrak{M}$ to Σ as $\mathfrak{M}|_\Sigma = (M, I \cap \{\sigma^{\mathfrak{M}} \mid \sigma \in \Sigma\})$. ■

On patterns in the restricted signature, model restrictions behave the same as the original model. This argument is important in our results, therefore we state it explicitly in the following proposition.

Proposition 2.2.2. Let $\mathfrak{M}$ be a Σ'-model and $\Sigma \subseteq \Sigma'$ be some signature. For every $\varphi \in \text{PATTERN}_\Sigma$ and every M-valuation ρ we have $\rho^{\mathfrak{M}}(\varphi) = \rho^{\mathfrak{M}|_\Sigma}(\varphi)$. ■

2.3 SYNTACTIC SUGAR

Example 2.2.1 already introduced the sugar $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$. Seeing how semantics work in ML, we can now define other expected syntactic sugar for the logical connectives such as "$\rightarrow$", "$\forall x.\, \varphi$", etc. Every time we write a pattern with syntactic sugar, we mean the desugared pattern.

$$\begin{aligned}
&\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2) && \forall x.\, \varphi \equiv \neg\exists x.\, \neg\varphi \\
&\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2 && \top \equiv (\exists x.\, x) \vee \neg(\exists x.\, x) \\
&\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1) && \bot \equiv \neg\top
\end{aligned}$$

Notice that $\top$ is a closed Σ-pattern with $\Sigma = \emptyset$ and if we replace $\exists x.\, x$ in $\top$ with a propositional variable p, then we get a propositional

3 Note that for every closed pattern φ we have $\rho|_{\mathsf{FV}(\varphi)} = \rho|_\emptyset = \emptyset$.

tautology $p \vee \neg p$. It is easy to see how $\top \equiv (\exists x.\, x) \vee \neg(\exists x.\, x)$ matches all elements of any model $\mathfrak{M}$. Let $\exists x.\, x$ evaluate to a set $A \subseteq M$, then $\top$ evaluates to $A \cup (M \setminus A) = M$. In fact, propositional tautologies always match all elements of a model:

Proposition 2.3.1 ([27]). Let ψ be a propositional tautology containing only variables $p_1, \ldots, p_n$ and let $\varphi_1, \ldots, \varphi_n \in \textsc{Pattern}_\Sigma$ for some signature $(\textsc{Var}, \Sigma)$. Then for every Σ-model $\mathfrak{M}$ and every M-valuation ρ we get that $\rho^{\mathfrak{M}}(\psi[\varphi_1/p_1, \ldots, \varphi_n/p_n]) = M$. ■

It is not hard to derive with basic set theory that the rest of the syntactic sugar works as expected too:

Proposition 2.3.2 ([27]). Let $\mathfrak{M}$ be a $(\textsc{Var}, \Sigma)$-model and ρ any M-valuation. The following propositions hold for all $x \in \textsc{Var}$ and all $\varphi \in \textsc{Pattern}_\Sigma$:

- $\rho^{\mathfrak{M}}(\top) = M$ and $\rho^{\mathfrak{M}}(\bot) = \emptyset$,
- $\rho^{\mathfrak{M}}(\varphi_1 \vee \varphi_2) = \rho^{\mathfrak{M}}(\varphi_1) \cup \rho^{\mathfrak{M}}(\varphi_2)$,
- $\rho^{\mathfrak{M}}(\varphi_1 \to \varphi_2) = (M \setminus \rho^{\mathfrak{M}}(\varphi_1)) \cup \rho^{\mathfrak{M}}(\varphi_2)$
  $= M \setminus (\rho^{\mathfrak{M}}(\varphi_1) \setminus \rho^{\mathfrak{M}}(\varphi_2))$,
- $\rho^{\mathfrak{M}}(\varphi_1 \leftrightarrow \varphi_2) = M \setminus (\rho^{\mathfrak{M}}(\varphi_1) \triangle \rho^{\mathfrak{M}}(\varphi_2))$,
- $\rho^{\mathfrak{M}}(\forall x.\, \varphi) = \bigcap_{m \in M} \rho[m/x]^{\mathfrak{M}}(\varphi)$,

where "$\triangle$" is set symmetric difference. ■

Notice that we defined the sugar $\forall x.\, \varphi$. We will extend this a little further and write $\forall \varphi$ to mean the *universal closure* of φ, i.e.,

$$\forall \varphi(x_1, \ldots, x_n) \equiv \forall x_1 \ldots \forall x_n.\, \varphi(x_1, \ldots, x_n).$$

This notation applied to sets of patterns will mean $\forall \Gamma \equiv \{\forall \gamma \mid \gamma \in \Gamma\}$.

2.4 ENTAILMENT

We are now ready to define the relation $\models_{\mathsf{ML}}$. Throughout the thesis, we usually drop ML in $\models_{\mathsf{ML}}$ and simply write $\models$. Because ML does not distinguish between formulas and terms, patterns play a dual role. We have seen that each pattern is interpreted as a set of elements given by the pattern valuations $\rho^{\mathfrak{M}}$ (Definition 2.2.2). Here we learn to think of patterns in their second role as formulas play in FOL, i.e., patterns can specify properties of models. A pattern is called *valid* in an ML model if the pattern matches *all* elements of the model, regardless of how we interpret variables.

Definition 2.4.1 (Validity). Let $\mathfrak{M}$ be a Σ-model. We say that a pattern $\varphi \in \textsc{Pattern}_\Sigma$ is *valid* in $\mathfrak{M}$, denoted $\mathfrak{M} \models \varphi$, iff $\rho^{\mathfrak{M}}(\varphi) = M$ for every M-valuation ρ. ■

We can also ask about validity of patterns w.r.t. sets of patterns, which we call *theories* if we assume a fixed signature. Matching logic theories play a role similar to FOL theories in that they specify models. A Σ-model $\mathfrak{M}$ is called a model *of* a Σ-theory Γ if all patterns of Γ are valid in $\mathfrak{M}$. Each pattern of a theory is called an *axiom* (of the theory).

Definition 2.4.2 (Theory). Let Σ be a signature. Any set $\Gamma \subseteq \textsc{Pattern}_\Sigma$ is called a Σ-theory (or simply theory when Σ is known from context). A *model of* Γ is any Σ-model $\mathfrak{M}$ such that $\mathfrak{M} \models \gamma$ for all $\gamma \in \Gamma$, in which case we write $\mathfrak{M} \models \Gamma$. We say φ is valid in Γ iff $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$ of Γ, in which case we write $\Gamma \models \varphi$. Finally, the pattern φ is simply *valid* if $\emptyset \models \varphi$. ■

Similar to FOL, adding universal quantification preserves validity in ML. That is why universal closure $\forall\varphi$ we defined in Section 2.3 makes good sense. Often we can suppose that a pattern is closed without loss of generality.

Lemma 2.4.1 ([10]). $\mathfrak{M} \models \varphi$ iff $\mathfrak{M} \models \forall x.\, \varphi$. ■

Corollary 2.4.1. $\Gamma \models \varphi$ iff $\Gamma \models \forall\varphi$. ■

Notice that finite theories can be understood as a single pattern, i.e., a conjunction of patterns. This follows straight from the definition of the relation $\models$.

Proposition 2.4.1. Let Γ_{fin} be a *finite* Σ-theory. Then $\mathfrak{M} \models \Gamma_{\text{fin}}$ iff $\mathfrak{M} \models \bigwedge \Gamma_{\text{fin}}$. ■

We call a theory Γ *satisfiable* iff there exists a model $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$. Because finite theories can be thought of as patterns (Proposition 2.4.1), in every model they evaluate to a set of elements. It might be surprising that unsatisfiable patterns (finite theories) do not always evaluate to $\emptyset$. A trivial example can be found by looking at the pattern $\neg x$:

Example 2.4.1. Consider the theory $\{\neg x\}$. Given any model $\mathfrak{M}$ with $|M| > 1$, for every M-valuation ρ it is easy to see that

$$\emptyset \subset \rho^{\mathfrak{M}}(\neg x) = M \setminus \{\rho(x)\} \subset M.$$

■

There is a new result stronger than Example 2.4.1 that we cover in Chapter 5.

A special class of patterns with nice semantics is called *predicate patterns*. The naming is not a coincidence; predicate patterns are always either true/valid (match all elements) or false (match *no* elements):

Definition 2.4.3 (Predicate pattern). Let $\mathfrak{M}$ be a Σ-model. A pattern $\psi \in \textsc{Pattern}_\Sigma$ is called an $\mathfrak{M}$-*predicate* (predicate in $\mathfrak{M}$) if

$$\text{for every } M\text{-valuation } \rho, \text{ either } \rho^{\mathfrak{M}}(\psi) = M \text{ or } \rho^{\mathfrak{M}}(\psi) = \emptyset.$$

If ψ is an $\mathfrak{M}$-predicate for every model $\mathfrak{M}$ of a Σ-theory Γ, then ψ is called a Γ-predicate (predicate in Γ). Finally, if ψ is an $\emptyset$-predicate (predicate in the empty theory $\emptyset$), then ψ is simply called a *predicate pattern*. ■

$\mathfrak{M}$-predicates have many of the properties that we would expect from FOL formulas. Namely, for any *closed* $\mathfrak{M}$-predicate ψ we have

$$\mathfrak{M} \models \psi \text{ iff } \mathfrak{M} \not\models \neg\psi.$$

The direction $(\Leftarrow)$ does not hold for arbitrary closed patterns. For example, in the Σ-model $\mathfrak{M} : M = \{0,1\}, \lambda^{\mathfrak{M}} = \{0\}$ we have both $\mathfrak{M} \not\models \neg\lambda$ and $\mathfrak{M} \not\models \lambda$ where $\lambda \in \Sigma_0$ is a closed pattern.

We can also prove that $\mathfrak{M}$-predicates are preserved under all standard connectives:

Proposition 2.4.2 ([27]). Let ψ_1, ψ_2 be $\mathfrak{M}$-predicates. Then $\neg\psi_1$, $\psi_1 \wedge \psi_2$, $\exists x.\, \psi_1$, $\forall x.\, \psi_1$ are all $\mathfrak{M}$-predicates. ■

Corollary 2.4.2. Let φ be a *closed* Γ-predicate. Then $\Gamma \models \varphi$ iff $\Gamma \cup \{\neg\varphi\} \models \bot$.

Proof.

- $(\Rightarrow)$ By contraposition. Let $\Gamma \cup \{\neg\varphi\} \not\models \bot$, thus there is some model $\mathfrak{M}$ of $\Gamma \cup \{\neg\varphi\}$ such that $\mathfrak{M} \not\models \bot$. In this model $\mathfrak{M}$, for every M-valuation ρ we have $\rho^{\mathfrak{M}}(\neg\varphi) = M$ by definition of $\models$. But then $\rho^{\mathfrak{M}}(\varphi) = \emptyset$: thus $\mathfrak{M} \models \Gamma$ and $\rho^{\mathfrak{M}}(\varphi) \neq M$, i.e., $\Gamma \not\models \varphi$.
- $(\Leftarrow)$ By contraposition. Let $\Gamma \not\models \varphi$, thus for some model $\mathfrak{M}$ of Γ we have $\mathfrak{M} \not\models \varphi$. This means $\rho^{\mathfrak{M}}(\varphi) = \emptyset$ for *all* M-valuations ρ because φ is a *closed* Γ-predicate. But then $\rho^{\mathfrak{M}}(\neg\varphi) = M$ for all M-valuations. By definition of $\models$, $\mathfrak{M} \models \Gamma \cup \{\neg\varphi\}$ and obviously $\mathfrak{M} \not\models \bot$ ($\rho^{\mathfrak{M}}(\emptyset) = \emptyset$). Thus $\Gamma \cup \{\neg\varphi\} \not\models \bot$.

■

One other important property to consider about Γ-predicates is that they are predicates also in all supersets of the original theory. This is simply because all models of the supersets are also models of the original theory.

Proposition 2.4.3. Let φ be a Γ-predicate and $\Gamma \subseteq \Gamma^+$. Then φ is also a Γ^+-predicate.

Proof. Immediate from $\mathfrak{M} \models \Gamma^+$ implying $\mathfrak{M} \models \Gamma$. ■

REMARK. Implications $\varphi_1 \to \varphi_2$ are not predicate patterns for all φ_1, φ_2, which has consequences that might be surprising. Let us assume that φ_1, φ_2 are some *closed* FOL formulas. In any FOL structure $\mathfrak{A}$, one is used to working with

$$\mathfrak{A} \models_{\mathsf{FOL}} \varphi_1 \to \varphi_2 \text{ iff } \mathfrak{A} \not\models_{\mathsf{FOL}} \varphi_1 \text{ or } \mathfrak{A} \models_{\mathsf{FOL}} \varphi_2.$$

This is fine because entailment in FOL is two-valued, i.e., φ_1 is either true or false in $\mathfrak{A}$. We have $\mathfrak{A} \models_{\mathsf{FOL}} \neg\varphi_1$ iff $\mathfrak{A} \not\models_{\mathsf{FOL}} \varphi_1$.

On the other hand, the ML implication $\varphi_1 \to \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$ is interpreted as set union, where $\neg\varphi_1$ can evaluate to any set of model elements. Recall that $\mathfrak{M} \not\models \varphi_1$ iff φ_1 does not match *all* elements. Thus for some models $\mathfrak{M}$ with $\emptyset \subset \rho^{\mathfrak{M}}(\varphi_1)$ we have $\rho^{\mathfrak{M}}(\neg\varphi_1) \subset M$, i.e.,

Our basic intuition about implication can mislead us.

$$\mathfrak{M} \not\models \varphi_1 \text{ does not imply } \mathfrak{M} \models \neg\varphi_1 \vee \varphi_2!$$

The implication $\varphi_1 \to \varphi_2$ can evaluate to any set of model elements because it is just a union of matched elements.

A better intuition for "$\to$" is given by pattern matching. An element matches $\varphi_1 \to \varphi_2$ iff it does not match φ_1 or matches φ_2. In other words, if an element matches φ_1 and $\varphi_1 \to \varphi_2$, then it must also match φ_2. This is an instance of modus ponens. Only if for *every* model element m holds that m matches φ_1 implies m matches φ_2, the pattern $\varphi_1 \to \varphi_2$ is valid. A valid ML implication corresponds to *subsumption*.

Proposition 2.4.4 ([27]). Let $\mathfrak{M}$ be a Σ-model. Then $\mathfrak{M} \models \varphi_1 \to \varphi_2$ iff for every M-valuation ρ we have $\rho^{\mathfrak{M}}(\varphi_1) \subseteq \rho^{\mathfrak{M}}(\varphi_2)$. ■

Thus the ML implication says something else than first-order implication. An analogue of first-order implication is discussed when we talk about the deduction property (Section 4.2.3).

2.5 EQUALITY AND DEFINEDNESS

We are looking for a pattern $\varphi_1 = \varphi_2$ that is valid in any model iff φ_1 and φ_2 match the same model elements. Formally, for such a pattern we would have

$$\mathfrak{M} \models \varphi_1 = \varphi_2 \text{ iff for every } M\text{-valuation } \rho^{\mathfrak{M}}(\varphi_1) = \rho^{\mathfrak{M}}(\varphi_2).$$

Is this possible without extending the syntax of ML? Note that FOL with equality is an extension of FOL in both syntax and semantics. There is indeed no FOL formula $\mathsf{EQ}(t_1, t_2)$ that is true in any given FOL interpretation iff the terms t_1, t_2 point to the same element of the FOL model. An easy way to prove this is using the Lowenheim-Skolem theorem (LST):

Theorem 2.5.1 ("Upward" Lowenheim-Skolem [18]). If a countable set of formulas without equality has a model, then it has a model of arbitrarily larger cardinality. ■

Corollary 2.5.1. Every satisfiable set of countably many FOL formulas has an infinite model. ■

Note that the version of LST we use here is for FOL without equality, and is rather trivial. The intuition is that having a model of a theory of FOL without equality, this model is non-empty and you can always add dummy elements that "behave" the same as some chosen original element of the model. For more details one can see, e.g. [18, p. 227].

Corollary 2.5.2. There is no FOL formula capturing equality of terms.

Proof. Suppose for a contradiction that there is some FOL formula $\mathsf{EQ}(t_1, t_2)$ such that for every FOL interpretation $(\mathfrak{A}, v)$

$$(\mathfrak{A}, v) \models_{\mathsf{FOL}} \mathsf{EQ}(t_1, t_2) \text{ iff } v^{\mathfrak{A}}(t_1) = v^{\mathfrak{A}}(t_2).$$

Then the theory $\{\forall x \forall y.\, \mathsf{EQ}(x, y)\}$ is countable and satisfiable but has no infinite model, which is a contradiction with Lowenheim-Skolem. ■

ML is expressive enough to capture equality of two patterns without any extensions to the logic itself. Because of Proposition 2.4.4, one might think that $\leftrightarrow$ does the job. Indeed, a corollary of Proposition 2.4.4 is the following:

Proposition 2.5.1 ([27]). Let $\mathfrak{M}$ be a Σ-model. Then $\mathfrak{M} \models \varphi_1 \leftrightarrow \varphi_2$ iff for every M-valuation ρ we have $\rho^{\mathfrak{M}}(\varphi_1) = \rho^{\mathfrak{M}}(\varphi_2)$. ■

Proposition 2.5.1 answers our question mentioned in the introduction of this section. Unfortunately, there are two problems with considering "$\leftrightarrow$" as equality. The first problem is that "$\leftrightarrow$" does not work as pattern equality when nested inside binders. A nice counterexample was provided in [27, p. 23]:

Example 2.5.1 ([27]). Suppose we want to model a unary symbol f as an FOL *term* (function). This means we want a pattern ψ such that for any model

$$\mathfrak{M} \models \psi \text{ iff } |f_M(m)| = 1 \text{ for every } m \in M.$$

If "$\leftrightarrow$" is pattern equality, the pattern $\psi \equiv \exists y.\, f(x) \leftrightarrow y$ should precisely capture this. However, consider the model

$$\mathfrak{M} : M = \{0, 1\}, f_M(0) = \emptyset, f_M(1) = \{0, 1\}.$$

Then for every M-valuation ρ we have

$$\begin{aligned} \rho^{\mathfrak{M}}(\exists y.\, f(x) \leftrightarrow y) &= \bigcup_{m \in M} M \setminus (f_M(\rho(x)) \triangle \{m\}) \\ &= (M \setminus \{0\}) \cup (M \setminus \{1\}) \\ &= M. \end{aligned}$$

This shows $\mathfrak{M} \models \exists y.\, f(x) \leftrightarrow y$ and clearly $|f_M(0)| \neq 1$. ■

There is also the second problem why $\leftrightarrow$ cannot be equality: $\varphi_1 \leftrightarrow \varphi_2$ is not a predicate pattern. In particular, $\rho^{\mathfrak{M}}(\varphi_1) \neq \rho^{\mathfrak{M}}(\varphi_2)$ does not imply $\rho^{\mathfrak{M}}(\varphi_1 \leftrightarrow \varphi_2) = \emptyset$. This is the actual cause of our bug in Example 2.5.1. The pattern $f(x) \leftrightarrow y$ matches some element for each valuation of y in the counterexample model $\mathfrak{M}$; the union of these elements contains every element in $\mathfrak{M}$.

However, notice $\varphi_1 \leftrightarrow \varphi_2$ is not that "far" from pattern equality. We only have to force $\varphi_1 \leftrightarrow \varphi_2$ to match the empty set if $\rho^{\mathfrak{M}}(\varphi_1) \neq \rho^{\mathfrak{M}}(\varphi_2)$. To do this, it turns out we only need to define a unary symbol $\lceil \cdot \rceil$ to behave as a ceiling function: $\lceil \varphi \rceil$ matches all elements iff φ matches at least one element. Then $\neg\lceil\neg(\varphi_1 \leftrightarrow \varphi_2)\rceil$ yields us exactly what we need. To see why, we can do an intuitive case analysis as follows.

- $\varphi_1 \leftrightarrow \varphi_2$ is valid: $\neg\lceil\neg(\varphi_1 \leftrightarrow \varphi_2)\rceil$ is valid because the negation $\neg(\varphi_1 \leftrightarrow \varphi_2)$ matches no element.
- $\varphi_1 \leftrightarrow \varphi_2$ is not valid: then the negation $\neg(\varphi_1 \leftrightarrow \varphi_2)$ matches at least one element. But then we assume $\lceil\neg(\varphi_1 \leftrightarrow \varphi_2)\rceil$ matches all elements, so $\neg\lceil\neg(\varphi_1 \leftrightarrow \varphi_2)\rceil$ is empty.

How can we define $\lceil \cdot \rceil$ to behave this way? It is not that difficult; we only need $\lceil \cdot \rceil^{\mathfrak{M}}(m) = M$ for all $m \in M$! How do we specify models with such a definition of $\lceil \cdot \rceil$? We consider theories containing the axiom $\lceil x \rceil$, which enforces the symbol $\lceil \cdot \rceil$ to behave this way:

Definition 2.5.1 (Definedness). Let Γ be a Σ-theory such that $\sigma(x) \in \Gamma$ for some unary $\sigma \in \Sigma_1$.[4] Then $\sigma(x)$ is called a *definedness axiom*. For $\sigma = \lceil \cdot \rceil$ we call $\lceil x \rceil$:

$$\text{(Definedness)} \qquad \lceil x \rceil.$$

For this particular symbol $\lceil \cdot \rceil$ we also define *totality* "$\lfloor \cdot \rfloor$", *equality* "$=$", *membership* "$\in$", and *set containment*[5] "$\subseteq$" as derived constructs:

$$\lfloor \varphi \rfloor \equiv \neg\lceil\neg\varphi\rceil \qquad \varphi_1 = \varphi_2 \equiv \lfloor \varphi_1 \leftrightarrow \varphi_2 \rfloor$$
$$x \in \varphi \equiv \lceil x \wedge \varphi \rceil \qquad \varphi_1 \subseteq \varphi_2 \equiv \lfloor \varphi_1 \rightarrow \varphi_2 \rfloor$$

■

To avoid writing too many parentheses, let us assume that all of the constructs in Definition 2.5.1 have the same precedence as symbol application, i.e., they bind more tightly than binary connectives. For example,

$$\neg\varphi_1 = \varphi_2 \wedge (x \in \neg\psi \rightarrow \varphi) \equiv ((\neg\varphi_1) = \varphi_2) \wedge ((x \in (\neg\psi)) \rightarrow \varphi).$$

The constructs from Definition 2.5.1 have their expected meaning. Specifically, in models $\mathfrak{M}$ satisfying the axiom $\lceil x \rceil$ these constructs

4 Note that this is the same as having $\forall x.\, \sigma(x) \in \Gamma$

5 Recall that $\rightarrow$ works as subsumption only in models where it is valid (Section 2.4)!

are $\mathfrak{M}$-predicates that decide whether a pattern φ matches at least one element ($\lceil\varphi\rceil$); matches all elements ($\lfloor\varphi\rfloor$); whether two patterns φ_1, φ_2 match the same elements ($\varphi_1 = \varphi_2$); etc. This is formally stated in the next proposition.

Proposition 2.5.2 ([27]). Let $\mathfrak{M}$ be a Σ-model such that $\mathfrak{M} \models \lceil x \rceil$. Then the following properties hold for every M-valuation ρ:

$$\rho^{\mathfrak{M}}(\lceil\varphi\rceil) = \begin{cases} M & \rho^{\mathfrak{M}}(\varphi) \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

$$\rho^{\mathfrak{M}}(\lfloor\varphi\rfloor) = \begin{cases} M & \rho^{\mathfrak{M}}(\varphi) = M \\ \emptyset & \text{otherwise} \end{cases}$$

$$\rho^{\mathfrak{M}}(\varphi_1 = \varphi_2) = \begin{cases} M & \rho^{\mathfrak{M}}(\varphi_1) = \rho^{\mathfrak{M}}(\varphi_2) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\rho^{\mathfrak{M}}(x \in \varphi) = \begin{cases} M & \rho(x) \in \rho^{\mathfrak{M}}(\varphi) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\rho^{\mathfrak{M}}(\varphi_1 \subseteq \varphi_2) = \begin{cases} M & \rho^{\mathfrak{M}}(\varphi_1) \subseteq \rho^{\mathfrak{M}}(\varphi_2) \\ \emptyset & \text{otherwise} \end{cases}$$

■

There is also another property of (Definedness) to consider. Because $\mathfrak{M}$-predicates are either true (evaluate to M) or false (evaluate to $\emptyset$) in the model $\mathfrak{M}$, it is easy to see that (Definedness) and totality are *identities* on $\mathfrak{M}$-predicates:

Proposition 2.5.3 ([27]). Let $\mathfrak{M} \models \lceil x \rceil$ and ψ be an $\mathfrak{M}$-predicate. Then for every $\mathfrak{M}$-valuation ρ we have $\rho^{\mathfrak{M}}(\lfloor\psi\rfloor) = \rho^{\mathfrak{M}}(\psi) = \rho^{\mathfrak{M}}(\lceil\psi\rceil)$. ■

Intuitively we can see that we fix Example 2.5.1 if we use $\exists y.\, \varphi = y$ instead of $\exists y.\, \varphi \leftrightarrow y$. How do we know that "=" meets our expectations about equality in other cases as well? One way how we can convince ourselves is to show that "=" behaves at least as good as equality in FOL with equality. How do we do this? We show that the equality axioms for FOL hold in ML.

Proposition 2.5.4 ([27]). Equality axioms for FOL are valid in ML theories containing $\lceil x \rceil$, i.e.:

(1) $\{\lceil x \rceil\} \models \varphi = \varphi$ (equality introduction),

(2) $\{\lceil x \rceil\} \models (\varphi_1 = \varphi_2 \wedge \varphi[\varphi_1/x]) \rightarrow \varphi[\varphi_2/x]$ (equality elimination).

■

Note that the equality elimination axiom is actually stronger in ML because the compared patterns φ_1, φ_2 do not have to be terms (in the FOL sense). We should also mention the axiom

$$\Big(\bigwedge_{1 \leq i \leq n} \varphi_i = \varphi_i' \Big) \to \sigma(\varphi_1, \ldots, \varphi_n) = \sigma(\varphi_1', \ldots, \varphi_n');$$

this axiom is valid in $\{\lceil x \rceil\}$, however, we cite a stronger result in Section 4.2.2. All remaining doubts about "=" should hopefully be settled in Chapter 3.

2.6 EQUALITY EXTENSIONS

As we have seen in Section 2.5, pattern equality $\varphi_1 = \varphi_2$ can be defined as an ML theory without any extensions to ML, only assuming one unary symbol $\lceil \cdot \rceil$ and the pattern (Definedness) $\equiv \lceil x \rceil$ as an axiom. We often consider theories containing (Definedness) because the axiom is special for many reasons. This axiom will follow us throughout the whole thesis.

Instead of mentioning whether a theory contains (Definedness), we will introduce a new concept called an *equality extension*. The equality extension of a theory Γ is Γ extended with a definedness axiom such that the added definedness axiom uses a *fresh* unary (definedness) symbol. We formalize equality extensions in the following definition.

Definition 2.6.1 (Equality extension). Let Γ be a Σ-theory. Then we call the $\Sigma_=$-theory $\Gamma_= \equiv \Gamma \cup \{\lfloor x \rfloor\}$ an *equality extension* of Γ where $\lfloor \cdot \rfloor \notin \Sigma$ and $\Sigma_= = \Sigma \cup \{\lfloor \cdot \rfloor\}$. For each signature Σ we assume some determined fresh symbol, i.e., for every pair of Σ-theories Γ, Γ' we assume $\Gamma_=, \Gamma'_=$ are $\Sigma_=$-theories such that $\Gamma_= \setminus \Gamma = \Gamma'_= \setminus \Gamma'$. ■

We must be careful. Observe that $(\cdot)_=$ is a *notation* that is undeterministic. There are infinitely many equality extensions of the same theory. To use the notation $\Gamma_=$ for a Σ-theory Γ, we have to agree on the meta-level that the fresh symbol is determined for Σ. This is w.l.o.g. because $(\cdot)_=$ is really only a shorthand. Instead of $\Gamma_=, \Gamma'_=$ we could always write: "Let Γ, Γ' be Σ-theories and $\lfloor \cdot \rfloor \notin \Sigma$ be some fresh unary symbol. Consider the theories $\Gamma \cup \{\lfloor x \rfloor\}$, $\Gamma' \cup \{\lfloor x \rfloor\}$."

We always have $\Gamma \subset \Gamma_=$ by construction and the added axiom $\lfloor x \rfloor$ is the only pattern in $\Gamma_=$ with the fresh symbol $\lfloor \cdot \rfloor$. Obviously we have a guarantee that $\Gamma_=$ contains a definedness axiom (with a fresh symbol), whereas Γ may or may not contain a definedness axiom (regardless of the used symbol). For example, the set

$$\{\lceil x \rceil\}_= \equiv \{\lceil x \rceil, \lfloor x \rfloor\}$$

is an equality extension of $\{\lceil x \rceil\}$. The fresh axiom is added even though $\{\lceil x \rceil\}$ already contains (Definedness).

Equality extensions are not an ad hoc concept: they make presentation of some existing results more compact and clear. Most importantly, they are crucial in our search for a complete proof system for matching logic (see Chapter 5). In Definition 2.6.1 we used $\lceil \cdot \rceil$ instead of $\lceil \cdot \rceil$ on purpose to make it more explicit that $\lceil \cdot \rceil$ is fresh. However, for simplicity we always assume w.l.o.g. that $\lceil x \rceil$ is the added (fresh) definedness axiom in $\Gamma_=$ unless stated otherwise, i.e., $\lceil x \rceil \in \Gamma_=$ and $\lceil x \rceil \notin \Gamma$.

3

CONNECTIONS WITH FIRST-ORDER LOGIC

We mention in the introduction that ML is an FOL variant. This is not an unjustified claim. In this chapter, we study how ML relates to FOL (and vice versa). We can translate patterns to first-order logic formulas and back while preserving the intended semantics. To prevent confusion, we will explicitly write $\models_{\mathsf{ML}}$ for ML semantics and $\models_{\mathsf{FOL}}$ for standard FOL semantics due to Tarski [30].

3.1 EMBEDDING ML IN FIRST-ORDER LOGIC WITH EQUALITY

We will show a sketch of the argument that every ML theory can be captured by predicate logic with equality, while preserving the original semantics. Predicate logic is a first-order logic fragment without function (or constant) symbols, i.e., first-order logic over relational symbol sets. It is known that FOL (even with equality) can be translated to a predicate logic with equality. The main idea is to consider the graph of a function instead of the function itself [13, p. 116].

For ML, we too can provide a direct translation to predicate logic with equality; there is indeed a good intuition why no function symbols are needed. The semantics of matching logic suggests that we can understand matching logic symbols as relations. For example, consider an ML model $\mathfrak{M}$ with a single n-ary symbol σ. We can define an $n+1$-ary relation R_σ such that

$$(m_1, \ldots, m_n, m) \in R_\sigma \text{ iff } m \in \sigma^{\mathfrak{M}}(m_1, \ldots, m_n).$$

Then $\mathfrak{M}$ can be translated to an FOL structure $\mathfrak{A}$ with a single predicate symbol σ interpreted as the relation R_σ, i.e., $\sigma^{\mathfrak{A}} := R_\sigma$! Let us turn our intuition into a method of translating patterns to FOL formulas. As [27] shows, we can define a translation of a pattern φ to an FOL formula $t(\varphi, m)$ that says: the pattern φ matches the element m.

Definition 3.1.1 ([27]). Let φ be a Σ-pattern. Then we define the translation $\overline{t}(\varphi) = \forall m.\, t(\varphi, m)$, where $t(\varphi, m)$ is a function defined inductively as follows:

- $t(x, m) = (x = m)$,
- $t(\neg\varphi, m) = \neg t(\varphi, m)$,
- $t(\varphi_1 \wedge \varphi_2, m) = t(\varphi_1, m) \wedge t(\varphi_2, m)$,
- $t(\exists x.\, \varphi, m) = \exists x.\, t(\varphi, m)$,

- $t(\sigma(\varphi_1, \ldots, \varphi_n), m) = \exists x_1 \ldots \exists x_n . P_\sigma(x_1, \ldots, x_n, m) \wedge \bigwedge_{1 \leq i \leq n} t(\varphi_i, x_i)$.

We further extend $\bar{t}$ to sets as $\bar{t}(\Gamma) = \{\bar{t}(\gamma) \mid \gamma \in \Gamma\}$. ■

Theorem 3.1.1 ([27]). Let Γ be a Σ-theory. Then for every Σ-pattern φ we have $\Gamma \models_{\mathsf{ML}} \varphi$ iff $\bar{t}(\Gamma) \models_{\mathsf{FOL}} \bar{t}(\varphi)$. ■

Note that we can use this translation to prove that our definition of "=" in ML directly translates to "=" as defined by predicate logic with equality [27, p. 51]. This can be seen as another piece of evidence that the ML meta-operator "=" does what it is meant to do.

3.2 EMBEDDING FIRST-ORDER LOGIC IN ML

First-order logic can be captured as an ML theory Γ_{FOL}, i.e., a theory for which we intuitively have

$$\models_{\mathsf{FOL}} \varphi \text{ iff } \Gamma_{\mathsf{FOL}} \models_{\mathsf{ML}} \varphi$$

for all FOL formulas φ. This was proved for many-sorted matching logic in [27, p. 38] constructing Γ_{FOL} with at least two sorts. Since we use a single-sorted variant of ML, we have to show a slightly different method sketched in [8, p. 6]. To mitigate reinventing the wheel, we include a full proof of a stronger result that for every FOL S-theory Φ there is an ML theory Γ_S such that

$$\Phi \models_{\mathsf{FOL}} \varphi \text{ iff } \Phi \cup \Gamma_S \models_{\mathsf{ML}} \varphi$$

for every FOL S-formula φ. This was proved neither in [27] nor [8]. Our construction has also nice properties that were not explicitly showed in [8].

Given an FOL signature $S = (\textsc{Var}, \textsc{Func}, \textsc{Pred})$ with *variables* $\textsc{Var}$, *function symbols* $\textsc{Func} = \{\textsc{Func}_0, \textsc{Func}_1, \ldots\}$ and *predicate symbols* $\textsc{Pred} = \{\textsc{Pred}_1, \textsc{Pred}_2, \ldots\}$, we can notice that FOL structures correspond to a special class of matching logic models if we allow FOL domains to be sets. Namely every FOL structure can be translated to an ML model as follows:

Definition 3.2.1. Let $S = (\textsc{Var}, \textsc{Func}, \textsc{Pred})$ be an FOL signature and $\mathfrak{A}$ be an FOL S-structure. Then we define an ML Σ_S-model $\mathfrak{M}$ corresponding to $\mathfrak{A}$ as

- $M = A$,
- $f^{\mathfrak{M}}(m_1, \ldots, m_n) = \{f^{\mathfrak{A}}(m_1, \ldots, m_n)\}$ for all $f \in \textsc{Func}_n$,
- $P^{\mathfrak{M}}(m_1, \ldots, m_n) = \begin{cases} M & \text{if } (m_1, \ldots, m_n) \in P^{\mathfrak{A}} \\ \emptyset & \text{otherwise} \end{cases}$ for all $P \in \textsc{Pred}_n$.

■

Given an FOL signature S, the idea is to construct a theory Γ_S of which every model corresponds to an FOL structure (and vice versa). The one-to-one correspondence is also why we choose $P^{\mathfrak{M}}$ in Definition 3.2.1 this way. How do we construct Γ_S? We want to force ML semantics to behave as FOL semantics. Recall that an ML symbol $\sigma \in \Sigma$ has relational semantics (Section 3.1). If we assume that (Definedness) is at our disposal, we can easily enforce a symbol σ to be functional (a *term*) with the following axiom [27]:

$$\text{(Function)} \qquad \exists y\,.\ \sigma(x_1, \ldots, x_n) = y$$

For the axiom (Function) we can show a lemma that confirms our intuition:

Lemma 3.2.1 ([27]). Let $\mathfrak{M}$ be a Σ-model and $\varphi \in \text{Pattern}_\Sigma$ be a pattern such that $y \notin \mathsf{FV}(\varphi)$. Then $\mathfrak{M} \models_{\mathsf{ML}} \exists y.\, \varphi = y$ iff for every M-valuation ρ there exists exactly one element $m \in M$ such that $\rho^{\mathfrak{M}}(\varphi) = \{m\}$. ■

Now our function axiom finally works.

To embed FOL syntax in ML, we define a matching logic signature (Var, Σ_S) where $\Sigma_S = \{\text{Func}_0\} \cup \{\Sigma_i \mid i \in \mathbb{N}^+, \Sigma_i = \text{Func}_i \cup \text{Pred}_i\}$. Notice that trivially every FOL S-formula is a Σ_S-pattern. We can define first-order logic in the signature S as an ML $\Sigma_S \cup \{\lceil \cdot \rceil\}$-theory Γ_S containing the (Function) axioms for all $f \in \text{Func}$ and axioms that enforce predicate symbols $P \in \text{Pred}$ to be Γ_S-predicates, i.e.,

$$\begin{aligned}\Gamma_S = &\{\lceil x \rceil\} \\ &\cup \{\exists y.\, f(x_1, \ldots, x_n) = y \mid f \in \text{Func}_n\}_{n \in \mathbb{N}} \\ &\cup \{P(x_1, \ldots, x_n) = \bot \vee P(x_1, \ldots, x_n) = \top \mid P \in \text{Pred}_n\}_{n \in \mathbb{N}^+}.\end{aligned}$$

Note that we w.l.o.g. assume that $\lceil \cdot \rceil \notin \text{Func} \cup \text{Pred}$. The following lemma shows that the axioms $\exists y.\, f(x_1, \ldots, x_n) = y$ suffice to enforce that all terms are singletons in a model of Γ_S, not just simple function applications:

Lemma 3.2.2 (Singleton terms [27]). Let $\mathfrak{M}$ be a Σ-model such that $\mathfrak{M} \models_{\mathsf{ML}} \lceil x \rceil$ and $\mathfrak{M} \models_{\mathsf{ML}} \exists y. f(x_1, \ldots, x_n) = y$ for all $f \in \text{Func}_n$ and all $n \in \mathbb{N}$. For every FOL term t in the signature $(\text{Var}, \text{Func}, \text{Pred})$ we have that $y \notin \mathsf{FV}(t)$ implies $\mathfrak{M} \models_{\mathsf{ML}} \exists y.\, t = y$.

More importantly, we can further show that any FOL S-formula φ is a Γ_S-predicate:

Lemma 3.2.3 (Γ_S-predicates). Let $S = (\text{Var}, \text{Func}, \text{Pred})$ be an FOL signature. Then every FOL S-formula φ is a Γ_S-predicate.

Proof. By structural induction on S-formula φ. The only non-trivial case is the base case.

- $\varphi \equiv P(t_1, \ldots, t_n)$: Let $\mathfrak{M}$ be any model of Γ_S and ρ be any M-valuation. By Lemma 3.2.2 $\mathfrak{M} \models_{\mathsf{ML}} \exists y.\, t_i = y$ for all $1 \leq i \leq n$. But then by definition $\rho^{\mathfrak{M}}(\exists y.\, t_i = y) = M$, which is by Lemma 3.2.1 iff there exists $m_i \in M$ such that $\rho^{\mathfrak{M}}(t_i) = \{m_i\}$ for all $1 \leq i \leq n$.

  We want $\rho^{\mathfrak{M}}(P(t_1, \ldots, t_n)) = \emptyset$ or $\rho^{\mathfrak{M}}(P(t_1, \ldots, t_n)) = M$. Consider some M-valuation ρ_x such that

  $$\rho_x(x_1) = m_1, \ldots, \rho_x(x_n) = m_n.$$

  Then we have

  $$\begin{aligned} \rho^{\mathfrak{M}}(P(t_1, \ldots, t_n)) &= P^{\mathfrak{M}}(\rho^{\mathfrak{M}}(t_1), \ldots, \rho^{\mathfrak{M}}(t_n)) \\ &= P^{\mathfrak{M}}(\rho_x^{\mathfrak{M}}(x_1), \ldots, \rho_x^{\mathfrak{M}}(x_n)) \\ &= \rho_x^{\mathfrak{M}}(P(x_1, \ldots, x_n)). \end{aligned}$$

  But $\rho_x^{\mathfrak{M}}(P(x_1, \ldots, x_n)) = \emptyset$ or $\rho_x^{\mathfrak{M}}(P(x_1, \ldots, x_n)) = M$ by construction. Because $\rho^{\mathfrak{M}}(P(t_1, \ldots, t_n)) = \rho_x^{\mathfrak{M}}(P(x_1, \ldots, x_n))$, we have what we wanted.

- Step. By IH and Proposition 2.4.2.

■

We can further argue that for S-formulas, Γ_S exactly captures standard FOL semantics $\models_{\mathsf{FOL}}$ due to Tarski [30]. This can be proved by showing that for every model of Γ_S, there is some corresponding FOL S-structure that behaves the same on S-formulas:

Definition 3.2.2. Let S be an FOL signature and $\mathfrak{M}$ be a model of Γ_S. We define a function $\mathfrak{M} \mapsto \mathfrak{A}$ where $\mathfrak{A}$ is an S-structure defined as follows:

- $A = M$,
- $f^{\mathfrak{A}}(m_1, \ldots, m_n) = m$ where $f^{\mathfrak{M}}(m_1, \ldots, m_n) = \{m\}$.
- $(m_1, \ldots, m_n) \in P^{\mathfrak{A}}$ if $P^{\mathfrak{M}}(m_1, \ldots, m_n) = M$,
  $(m_1, \ldots, m_n) \notin P^{\mathfrak{A}}$ if $P^{\mathfrak{M}}(m_1, \ldots, m_n) = \emptyset$ where $P \neq \lceil \cdot \rceil$.

■

Note that the function $\mathfrak{M} \mapsto \mathfrak{A}$ is well-defined because of Lemma 3.2.2 and Lemma 3.2.3. Simply choose a valuation that assigns $\rho(x_i) = m_i$, then $f(x_1, \ldots, x_n)$ must be a term, i.e., there exists $m \in M$ such that $\rho^{\mathfrak{M}}(f(x_1, \ldots, x_n)) = \{m\} = f^{\mathfrak{M}}(m_1, \ldots, m_n)$. There can only be one such m because $f^{\mathfrak{M}}$ is a function. Analogously for the predicate symbols (note that always $M \neq \emptyset$). We immediately notice that $\mathfrak{M} \mapsto \mathfrak{A}$ from Definition 3.2.2 maps ML models to such FOL models that behave the same on FOL formulas:

Theorem 3.2.1 (Isomorphism with FOL). Let S be an FOL signature. There exists a bijection $h : \{\mathfrak{M} \mid \mathfrak{M} \models_{\mathsf{ML}} \Gamma_S\} \to \{\mathfrak{A} \mid \mathfrak{A} \models_{\mathsf{FOL}} \emptyset\}$ such that

- for every FOL S-term t and valuation $v : \textsc{Var} \to A$ it holds that $\{v^{h(\mathfrak{M})}(t)\} = \rho^{\mathfrak{M}}(t)$ for the M-valuation $\rho := v$,
- for every FOL S-formula φ holds $h(\mathfrak{M}) \models_{\mathsf{FOL}} \varphi$ iff $\mathfrak{M} \models_{\mathsf{ML}} \varphi$.

Proof. Let us consider the function

$$h : \{\mathfrak{M} \mid \mathfrak{M} \models_{\mathsf{ML}} \Gamma_S\} \to \{\mathfrak{A} \mid \mathfrak{A} \models_{\mathsf{FOL}} \emptyset\}$$

given by Definition 3.2.2 and prove h is a bijection. Of course, h is injective by construction. For surjectivity, consider for a contradiction that there exists some FOL structure $\mathfrak{A}$ that is not covered. Use Definition 3.2.1 to construct an ML model $\mathfrak{M}$, extend $\mathfrak{M}$ with the (fresh) symbol $\lceil\cdot\rceil$ such that $\lceil\cdot\rceil^{\mathfrak{M}}(m) = M$ for all $m \in M$. Obviously $\mathfrak{M} \models_{\mathsf{ML}} \Gamma_s$ and $h(\mathfrak{M}) = \mathfrak{A}$, contradiction.

Consider any pair $(\mathfrak{M}, h(\mathfrak{M}))$ and set $\mathfrak{A} = h(\mathfrak{M})$. Let us now prove the first requirement on h by structural induction over S-terms.

- $t \equiv x$: $\{v^{\mathfrak{A}}(t)\} = \{v(x)\} = \{\rho(x)\} = \rho^{\mathfrak{M}}(x)$.
- $t \equiv f(t_1, \ldots, t_n)$: by IH $\{v^{\mathfrak{A}}(t_i)\} = \rho^{\mathfrak{M}}(t_i)$ for every $1 \leq i \leq t$. Thus

$$\begin{aligned}
\{v^{\mathfrak{A}}(f(t_1, \ldots, t_n))\} &= \{f^{\mathfrak{A}}(v^{\mathfrak{A}}(t_1), \ldots, v^{\mathfrak{A}}(t_n))\} \\
&\overset{h}{=} f^{\mathfrak{M}}(v^{\mathfrak{A}}(t_1), \ldots, v^{\mathfrak{A}}(t_n)) \\
&= f^{\mathfrak{M}}(\{v^{\mathfrak{A}}(t_1)\}, \ldots, \{v^{\mathfrak{A}}(t_n)\}) \\
&\overset{\text{IH}}{=} f^{\mathfrak{M}}(\rho^{\mathfrak{M}}(t_1), \ldots, \rho^{\mathfrak{M}}(t_n)) \\
&= \rho^{\mathfrak{M}}(f(t_1, \ldots, t_n))
\end{aligned}$$

Let us now prove the second requirement on h. We prove a stronger claim by structural induction on every S-formula φ that for every FOL interpretation $(\mathfrak{A}, v)$ we have

$$(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi \text{ iff } \rho^{\mathfrak{M}}(\varphi) = M \text{ for } \rho := v.$$

The only non-trivial case is the base case and negation.

- $\varphi \equiv P(t_1, \ldots, t_n)$.

$$\begin{aligned}
(\mathfrak{A}, v) \models_{\mathsf{FOL}} P(t_1, \ldots, t_n) \;&\text{ iff } (v^{\mathfrak{A}}(t_1), \ldots, v^{\mathfrak{A}}(t_n)) \in P^{\mathfrak{A}} \\
&\overset{h}{\text{iff}}\; P^{\mathfrak{M}}(v^{\mathfrak{A}}(t_1), \ldots, v^{\mathfrak{A}}(t_n)) = M \\
&\text{iff } P^{\mathfrak{M}}(\{v^{\mathfrak{A}}(t_1)\}, \ldots, \{v^{\mathfrak{A}}(t_n)\}) = M \\
&\overset{1}{\text{iff}}\; P^{\mathfrak{M}}(\rho^{\mathfrak{M}}(t_1), \ldots, \rho^{\mathfrak{M}}(t_n)) = M \\
&\text{iff } \rho^{\mathfrak{M}}(P(t_1, \ldots, t_n)) = M
\end{aligned}$$

  where (1) follows from the first requirement on h.

- $\varphi \equiv \neg\psi$. Then $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \neg\psi$ iff $(\mathfrak{A}, v) \not\models_{\mathsf{FOL}} \psi$ iff $\rho^{\mathfrak{M}}(\psi) \neq M$ for $\rho := v$. Consider that $\mathfrak{M} \models_{\mathsf{ML}} \Gamma_S$, thus ψ is an $\mathfrak{M}$-predicate. But then this is iff $\rho^{\mathfrak{M}}(\psi) = \emptyset$ iff $\rho^{\mathfrak{M}}(\neg\psi) = M$.

- $\varphi \equiv \psi_1 \wedge \psi_2$. Then $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi_1 \wedge \varphi_2$ iff $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi_1$ and $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \varphi_2$ iff $\rho^{\mathfrak{M}}(\varphi_1) = M$ and $\rho^{\mathfrak{M}}(\varphi_2) = M$ iff $\rho^{\mathfrak{M}}(\varphi_1 \wedge \varphi_2) = M$.

- $\varphi \equiv \exists x.\, \psi$. Then $(\mathfrak{A}, v) \models_{\mathsf{FOL}} \exists x.\, \psi$ iff $(\mathfrak{A}, v[m/x]) \models_{\mathsf{FOL}} \psi$ for some $m \in M$ iff $\rho[m/x]^{\mathfrak{M}}(\psi) = M$ for some $m \in M$ iff $\rho^{\mathfrak{M}}(\exists x.\, \psi) = \bigcup_{m \in M} \rho[m/x]^{\mathfrak{M}}(\psi) = M$. Note that we can use IH here because the statement is proved for *all* FOL interpretations $(\mathfrak{A}, v)$.

This yields what we set out to prove as $\mathfrak{A} \models_{\mathsf{ML}} \varphi$ iff $(\mathfrak{A}, v) \models_{\mathsf{ML}} \varphi$ for every valuation $v : \textsc{Var} \to A$ iff $\rho^{\mathfrak{M}}(\varphi) = M$ for every M-valuation $\rho : \textsc{Var} \to M$ iff $\mathfrak{M} \models_{\mathsf{ML}} \varphi$. Equivalence (2) is given by the proved statement and the fact that obviously $\{v \mid v : \textsc{Var} \to A\} = \{\rho \mid \rho : \textsc{Var} \to M\}$ because $A = M$ by construction. ■

Of course, Γ_S is satisfiable given any FOL signature S. We prove a stronger result that Γ_S is satisfiable even if it is in union with a satisfiable FOL theory.

Lemma 3.2.4. Let $S = (\textsc{Var}, \textsc{Func}, \textsc{Pred})$ be an FOL signature. Then for every FOL S-theory Φ we have that Φ is satisfiable in an FOL structure iff $\Phi \cup \Gamma_S$ is satisfiable in an ML model.

Proof.
($\Rightarrow$) Let $\mathfrak{A} \models_{\mathsf{FOL}} \Phi$. Take the corresponding ML model $\mathfrak{M}$ from Theorem 3.2.1, which yields that $\mathfrak{M} \models_{\mathsf{ML}} \Phi$. We also have $\mathfrak{M} \models_{\mathsf{ML}} \Gamma_S$ by construction, i.e., $\mathfrak{M} \models_{\mathsf{ML}} \Phi \cup \Gamma_S$.
($\Leftarrow$) Let $\mathfrak{M} \models_{\mathsf{ML}} \Phi \cup \Gamma_S$. Take the corresponding FOL structure $\mathfrak{A}$ from Theorem 3.2.1. Then $\mathfrak{A} \models_{\mathsf{FOL}} \Phi$. ■

Now we are ready to prove the main result of this section.

Theorem 3.2.2. Let S be an FOL signature and Φ some S-theory. Then for every S-formula φ we have $\Phi \models_{\mathsf{FOL}} \varphi$ iff $\Phi \cup \Gamma_S \models_{\mathsf{ML}} \varphi$.

Proof. It suffices to prove $\Phi \models_{\mathsf{FOL}} \forall\varphi$ iff $\Phi \cup \Gamma_S \models_{\mathsf{ML}} \forall\varphi$ by Lemma 2.4.1. Let $\bot_{\mathsf{FOL}}$ be some FOL contradiction. Because we know that φ is a Γ_S-predicate, by Corollary 2.4.2 it further suffices to prove

$$\Phi \cup \{\neg\forall\varphi\} \models_{\mathsf{FOL}} \bot_{\mathsf{FOL}} \text{ iff } \Gamma_S \cup \Phi \cup \{\neg\forall\varphi\} \models_{\mathsf{ML}} \bot.$$

This is immediate from Lemma 3.2.4 because $\Phi \cup \{\neg\forall\varphi\}$ is an FOL S-theory. $\Phi \cup \{\neg\forall\varphi\} \models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$ iff $\Phi \cup \{\neg\forall\varphi\}$ is not satisfiable iff $\Gamma_S \cup (\Phi \cup \{\neg\forall\varphi\})$ is not satisfiable iff $\Phi \cup \{\neg\forall\varphi\} \models_{\mathsf{ML}} \bot$. ■

4

TWO PROOF SYSTEMS FOR MATCHING LOGIC

We have defined in Section 2.2 a notion of truth and now we would like a tool that can mechanically *verify* truth, i.e., a proof system for matching logic. Chapter 2 already pointed out that ML fulfills intuitive preconditions for an elegant proof system. Propositional tautologies are valid (Proposition 2.3.1) and semantics of $\varphi_1 \to \varphi_2$ agree with modus ponens (Proposition 2.4.4). That means a proof system of matching logic can (and should) include a complete proof system for propositional logic. Since ML is a variant of FOL (Chapter 3), why not take a proof system of FOL as well?

This chapter covers two existing proof systems for ML. Section 4.1 covers a system called System $\mathcal{P}$ [27], which is based on the intuition of taking a complete FOL proof system and turning it into a proof system for matching logic. We shall see that System $\mathcal{P}$ stumbles when dealing with term substitutions $(\forall x.\, \varphi) \to \varphi[t/x]$. The workaround is painful; $\mathcal{P}$ directly depends on (Definedness) and the particular symbol $\lceil \cdot \rceil$, which means we cannot verify truth using $\mathcal{P}$ in all theories. Section 4.2 covers a system called System $\mathcal{H}$ [11], which is a successful attempt at a proof system that does not depend on any fixed formal symbols. What is more, $\mathcal{H}$ has numerous interesting properties that make it a very strong and flexible system for matching logic. Most importantly, $\mathcal{H}$ is well-suited for intended applications of ML.

4.1 SYSTEM P

A Hilbert-style proof system for ML has been known since the introduction of ML [27, p. 53]. Because ML is very close to FOL, the idea was to take a complete proof system for FOL and make it work for matching logic. However, there is a catch; we cannot use the following axiom for term substitutions:

$$(\forall x.\, \varphi) \to \varphi[t/x].$$

Term substitutions are problematic in matching logic exactly because terms are not syntactically distinguished from formulas. We cannot allow $(\forall x.\, \varphi) \to \varphi[\psi/x]$ for any pattern ψ because ψ can match any number of elements, not just one (as variables do). Take the counterexample pointed out by [27, p. 52], where we pick the pattern $\varphi \equiv (\exists y.\, x = y)$. Consider $\psi \equiv \neg x$, then for every Σ-model $\mathfrak{M}$ with $|M| \neq 2$ we have

$$\mathfrak{M} \not\models (\forall x.\, \exists y.\, x = y) \to (\exists y.\, (\neg x) = y).$$

(PT)	φ if φ is a propositional tautology over patterns
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
($\forall$)	$(\forall x.\, \varphi_1 \to \varphi_2) \to (\varphi_1 \to \forall x.\, \varphi_2)$ if $x \notin \mathsf{FV}(\varphi_1)$
(FunSub)	$((\exists y.\, \psi = y) \wedge \forall x.\, \varphi) \to \varphi[\psi/x]$ if $y \notin \mathsf{FV}(\psi)$
(Gen)	$\dfrac{\varphi}{\forall x.\, \varphi}$
(EqIntro)	$\varphi = \varphi$
(EqElim)	$(\varphi_1 = \varphi_2 \wedge \psi[\varphi_1/x]) \to \psi[\varphi_2/x]$
(MemIntro)	$\dfrac{\varphi}{\forall x.\, x \in \varphi}$ if $x \notin \mathsf{FV}(\varphi)$
(MemElim)	$\dfrac{\forall x.\, x \in \varphi}{\varphi}$ if $x \notin \mathsf{FV}(\varphi)$
(MemVar)	$(x \in y) = (x = y)$
(Mem$_\neg$)	$(x \in \neg\varphi) = \neg(x \in \varphi)$
(Mem$_\wedge$)	$(x \in \varphi_1 \wedge \varphi_2) = (x \in \varphi_1) \wedge (x \in \varphi_2)$
(Mem$_\exists$)	$(x \in \exists y.\, \varphi) = \exists y.(x \in \varphi)$ where x and y distinct.
(MemSym)	$(x \in C_\sigma[\varphi]) = \exists y.(y \in \varphi) \wedge (x \in C_\sigma[y])$ if $y \notin \mathsf{FV}\,(C_\sigma[\varphi])$

Figure 4.1: System $\mathcal{P}$

Intuitively, the workaround is to allow substituting ψ for which we can prove $\{\lceil x \rceil\} \models \exists y.\, \psi = y$, i.e., where ψ behaves as a *functional pattern*. This type of substitution is called *functional substitution* (FunSub) and can be shown to be valid, i.e.,

$$\{\lceil x \rceil\} \models ((\exists y.\, \psi = y) \wedge \forall x.\, \varphi) \to \varphi[\psi/x] \text{ if } y \notin \mathsf{FV}(\psi).$$

Building upon this idea, [27] presents System $\mathcal{P}$ that is sound and complete for theories containing (Definedness). This system can be divided into two groups of rules, as outlined by the separator in Figure 4.1. The first group is a proof system for predicate logic[1] with equality with a workaround for term substitutions. The second group are technical rules for "$\in$" that were needed to show completeness using a translation "backwards" from the translation in Section 3.1. The symbol C_σ occurring in the last rule will be explained in Section 4.2. For the rest of the details, we recommend interested readers to see [27, p. 54].

System $\mathcal{P}$ certainly serves its purpose and is educational by showing how matching logic proofs relate to FOL proofs; the proof of completeness goes by reduction to a complete FOL proof system. Unfortunately, the inspiration for $\mathcal{P}$ is also the very reason why $\mathcal{P}$ makes sense only

1 Predicate logic is first-order logic without functional symbols.

for theories containing (DEFINEDNESS).[2] System $\mathcal{P}$ directly contains the predefined constructs we derived with $\lceil\cdot\rceil$ in Section 2.5.

Theorem 4.1.1 ([27]). Let Γ be a Σ-theory containing (DEFINEDNESS). Then for every $\varphi \in \text{PATTERN}_\Sigma$ we have $\Gamma \models \varphi$ iff $\Gamma \vdash_{\mathcal{P}} \varphi$. ■

Observe that symbols such as "$=$" or "$\in$" are mere syntactic sugar over the fixed symbol $\lceil\cdot\rceil$, e.g., $\varphi_1 = \varphi_2 \equiv \neg\lceil\neg\varphi_1 \leftrightarrow \varphi_2\rceil$. What if Γ uses the symbol $\lceil\cdot\rceil$ for something else than (DEFINEDNESS), e.g., for $\neg\forall x.\, \lceil x\rceil$? System $\mathcal{P}$ does not make sense for these theories.

We defined the so-called *equality extensions* in Definition 2.6.1, which add a definedness axiom with a fresh symbol that can be different from $\lceil\cdot\rceil$. Why cannot System $\mathcal{P}$ use "$=$" that is a sugar for

$$\varphi_1 = \varphi_2 \equiv \neg\lfloor\neg\varphi_1 \leftrightarrow \varphi_2\rfloor?$$

We can define a class of proof systems that are the same as $\mathcal{P}$, except they use any definedness symbol we choose:

Definition 4.1.1 (System $\mathcal{P}_\Sigma$). Let Σ be some symbol set. Then by $\mathcal{P}_\Sigma$ we mean System $\mathcal{P}$ where "$=$" and "$\in$" use the fresh definedness symbol not in Σ that we assume to be determined. ■

Whatever fresh definedness symbol we agree to use for the notation $\Gamma_=$ given that Γ is a Σ-theory, we can take the same fresh definedness symbol, plug it into $\mathcal{P}$, call it $\mathcal{P}_\Sigma$ and from Theorem 4.1.1 we trivially know that $\mathcal{P}_\Sigma$ will be sound and complete w.r.t. the theory $\Gamma_=$:

[27] does not consider equality extensions but this result follows trivially from the original results.

Proposition 4.1.1. Let Γ be a Σ-theory Then for every $\varphi \in \text{PATTERN}_{\Sigma_=}$ we have $\Gamma_= \models \varphi$ iff $\Gamma_= \vdash_{\mathcal{P}_\Sigma} \varphi$.

Proof. We know that there is a definedness axiom in $\Gamma_=$, w.l.o.g. $\Gamma_= \setminus \Gamma = \{\lceil x\rceil\}$. This axiom uses the same (fresh) symbol as the one used in $\mathcal{P}_\Sigma$ by construction. Thus the proof is analogous to that of Theorem 4.1.1. ■

If we take $\mathcal{P}_{(\cdot)}$ as a family of proof systems and agree on some determined fresh symbols for each Σ, we could intuitively say that $\mathcal{P}_{(\cdot)}$ are sound and complete w.r.t. equality extensions $\Gamma_=$, even if Γ uses the "original" symbol $\lceil\cdot\rceil$ for something else than (DEFINEDNESS). Each System $\mathcal{P}_\Sigma$ makes sense for the equality extension of a theory in the signature Σ.

However, we are starting to see that the situation is not ideal. We do not want to redefine $\mathcal{P}$ if we want to use $\lceil\cdot\rceil$ for something else. Even if we forbid $\lceil\cdot\rceil$ a different meaning, Theorem 4.1.1 does not mean that $\mathcal{P}$ is complete. Not all theories contain (DEFINEDNESS). Even for as simple theory as $\{\forall x.\, x\}$, we do not know whether $\{\forall x.\, x\} \models \varphi$ implies $\{\forall x.\, x\} \vdash \varphi$ for all φ (we solve this instance in Section 5.1).

2 System $\mathcal{P}$ actually makes sense for every theory Γ such that $\Gamma \vdash \lceil x\rceil$.

One could suggest that we simply add $\lceil \cdot \rceil$ to the ML syntax with the expected semantics and add $\lceil x \rceil$ as an axiom to $\mathcal{P}$. However, this is contrary to the principles of ML. ML tries to build upon as small a core as possible because it is meant to be flexible and simple so that it is trustworthy.

We would naturally like to find a complete proof system for all theories or find a fundamental reason why such a system cannot exist. A counterexample would be interesting as matching logic can be easily embedded in FOL (Chapter 3) and for FOL we have a complete proof system. Apart from practical motivations, there is also a strictly theoretical one: the connection of ML to modal logic (see, e.g., [11]). If we find a proof system for ML without (Definedness), we might find an alternative proof system for several modal logics, which could be defined as ML theories. This would be another strong argument for ML as a logic unifying other logics.

4.2 System H

We have seen that System $\mathcal{P}$ (Section 4.1) makes sense only for theories containing (Definedness). Moreover, $\mathcal{P}$ is not well-suited for the intentions behind ML, which is discussed already in [27, pp. 53, 57]. The second group of rules in $\mathcal{P}$ are more technical than practical; they axiomatize working with the membership constructs "$\in$". Instead, we would like to axiomatize something more fundamental for matching logic and derive all of the technical rules as lemmas.

(PT)	φ if φ is a propositional tautology over patterns
(MP)	$\dfrac{\varphi_1 \quad \varphi_1 \to \varphi_2}{\varphi_2}$
($\forall$)	$(\forall x.\, \varphi_1 \to \varphi_2) \to (\varphi_1 \to \forall x.\, \varphi_2)$ if $x \notin \mathsf{FV}(\varphi_1)$
(Sub)	$(\forall x.\, \varphi) \to \varphi[y/x]$
(Gen)	$\dfrac{\varphi}{\forall x.\, \varphi}$
(Propagation$_\bot$)	$C_\sigma[\bot] \to \bot$
(Propagation$_\vee$)	$C_\sigma[\varphi_1 \vee \varphi_2] \to (C_\sigma[\varphi_1] \vee C_\sigma[\varphi_2])$
(Propagation$_\exists$)	$C_\sigma[\exists x.\varphi] \to \exists x.C_\sigma[\varphi]$ if $x \notin \mathsf{FV}(C_\sigma[\exists x.\varphi])$
(Framing)	$\dfrac{\varphi_1 \to \varphi_2}{C_\sigma[\varphi_1] \to C_\sigma[\varphi_2]}$ where C_σ is a single symbol context.
(Ex)	$\exists x.\, x$
(Singleton)	$\neg(C_1[x \wedge \varphi] \wedge C_2[x \wedge \neg\varphi])$ where C_1, C_2 are nested symbol contexts.

Figure 4.2: System $\mathcal{H}$

[10, 11] introduces System $\mathcal{H}$ (Figure 4.2)[3]; a Hilbert-style proof system for matching logic that does not depend on any formal symbols. Throughout the thesis, we drop $\mathcal{H}$ in $\vdash_{\mathcal{H}}$ and write just $\vdash$ instead. Let Γ be a Σ-theory. Then $\Gamma \vdash \varphi$ for $\varphi \in \textsc{Pattern}_\Sigma$ means there exists a Hilbert-style proof of φ in $\mathcal{H}$ with additional axioms from Γ. Of course, formulas of proof rules in $\mathcal{H}$ are over the same signature as Γ, i.e., Σ. In particular, System $\mathcal{H}$ does not contain any fixed predefined symbols. Instead, it works with so-called *contexts* (the notation C_σ).

Contexts are patterns with a distinguished free variable $\Box$ that serves as a *placeholder*. It is called a context because the placeholder can be substituted for any pattern without implicit α-renaming. For example, if

$$C_1 \equiv \sigma(\top) \wedge \forall x.\, \Box,$$

then we write $C_1[x]$ to mean $\sigma(\top) \wedge \forall x.\, x$. Intuitively, a context C is a symbol context iff the "path" to the free variable $\Box$ contains only symbols and no logical connectives. The context

$$C_2 \equiv \sigma_1(\sigma_1(\top), x \wedge \neg y, \sigma_2(\sigma_3(\Box), \lambda)),$$

is a symbol context because the path $\sigma_1, \sigma_2, \sigma_3$ to $\Box$ contains only symbols. On other other hand, $\sigma(\Box) \wedge \lambda$ is *not* a symbol context; the path to $\Box$ is $\wedge, \sigma$. Contexts are formally defined inductively in the next definition.

Definition 4.2.1 (Context). A pattern C is called a *context* if C contains a distinguished (occurring only once) free variable $\Box$. Then $C[\varphi]$ means the result of replacing $\Box$ with φ without implicit α-renaming.[4] Given $\sigma \in \Sigma_n$, a *single symbol context* C_σ is a context of the form

$$C_\sigma \equiv \sigma(\varphi_1, \ldots, \varphi_{i-1}, \Box, \varphi_{i+1}, \ldots, \varphi_n)$$

where $\varphi_1, \ldots, \varphi_n$ are patterns not containing $\Box$. A *nested symbol context* is defined inductively as follows.

- The variable $\Box$ is a nested symbol context.
- If C_σ is a single symbol context and C is a nested symbol context, then $C_\sigma[C]$ is a nested symbol context.

A nested symbol context is sometimes simply called a *symbol context*. ■

3 We work with the "$\forall$-version" of $\mathcal{H}$, which was actually used in [10].

4 Free variables in φ may be bound in $C[\varphi]$, which is different from capture-avoiding substitution.

PROOF RULES. System $\mathcal{H}$ is divided into four groups of rules as shown by the separators (Figure 4.2).

(1) The first group is a proof system for propositional logic, which can be include because of Proposition 2.3.1. Given a tautology of propositional logic such as $p \vee \neg p$, (PT) says that replacing each propositional variable p_i with a pattern φ_i in this propositional tautology yields an axiom. If we wish to avoid the meta-rule (PT) that adds infinitely many rules, we can simply replace it with any sound and complete system for propositional logic.

(2) The second group is a first-order proof system. A notable difference is that (Sub) is weaker than term substitution in FOL, as it only allows to substitute variables for variables. Notice that except for term substitution, all axiom schemes and proof rules of a complete proof system for FOL (without equality) are included in $\mathcal{H}$.

(3) The third group allows us to do *frame reasoning*, which is motivated by formal verification of data structures. We discuss these rules in Section 4.2.1.

(4) The fourth group consists of technical axioms.

Even though symbol contexts are an interesting concept, they are not special in the proof-theoretic sense. [10] gives a conventional proof that $\mathcal{H}$ is sound. Unlike $\mathcal{P}$, System $\mathcal{H}$ is sound for all theories even if they do not contain (Definedness). This is important because it makes sense to use it for any theory (unlike $\mathcal{P}$).

Theorem 4.2.1 (Soundness [10]). For every Σ-theory Γ and every pattern $\varphi \in \text{Pattern}_\Sigma$ we have that $\Gamma \vdash \varphi$ implies $\Gamma \models \varphi$. ■

CLOSED PATTERNS. Notice that (Sub) and (Gen) mean that we can focus only on closed patterns in the next discussion without loss of generality. Intuitively this is because we can always close a pattern with the corresponding universal quantifiers and vice versa. This is formally stated in the next theorem.

Proposition 4.2.1. Let Γ be a Σ-theory. Then for every $\varphi \in \text{Pattern}_\Sigma$ we have $\Gamma \vdash \varphi$ iff $\forall\Gamma \vdash \forall\varphi$.

Proof.
($\Rightarrow$) Let $\Gamma \vdash \varphi$. By definition of $\vdash$ there is a finite Σ-theory $\Gamma' \subseteq \Gamma$ such that $\Gamma' \vdash \varphi$. The following is the proof of $\forall\Gamma \vdash \forall\varphi$.

(1) Introduce each axiom from $\forall\Gamma' \subseteq \forall\Gamma$.

(2) For each introduced axiom $\forall\gamma \in \forall\Gamma'$ where $\gamma \in \Gamma'$, repeatedly apply (Sub) on $\forall\gamma$ to get γ.

(3) Apply the proof of $\Gamma' \vdash \varphi$.

(4) Finitely many times apply (GEN) to get $\forall\varphi$.

($\Leftarrow$) Symmetrically using (GEN) on free variables for each $\gamma \in \Gamma$. ■

CONDITIONAL COMPLETENESS. Much like System $\mathcal{P}$, System $\mathcal{H}$ is complete w.r.t. theories containing (DEFINEDNESS). This is because (MP) and (GEN) are in both $\mathcal{P}$ and $\mathcal{H}$ while for every axiom scheme α of $\mathcal{P}$ we have $\{\lceil x \rceil\} \vdash \alpha$ [10]. Of course, the situation is the same if we consider $\mathcal{P}$ defined with a different definedness symbol such as $\lceil\cdot\rceil$, e.g., $\mathcal{P}_\Sigma$ defined for equality extensions we assume determined for Σ. That is why soundness and "conditional" completeness of $\mathcal{H}$ can be expressed for any equality extension as in Proposition 4.2.2.

Theorem 4.2.2 ([10]). Let Γ be a Σ-theory containing (DEFINEDNESS). Then for every $\varphi \in \text{PATTERN}_\Sigma$ we have $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$. ■

Proposition 4.2.2. Let Γ be a Σ-theory. Then for every $\varphi \in \text{PATTERN}_{\Sigma_=}$ we have $\Gamma_= \models \varphi$ iff $\Gamma_= \vdash \varphi$. ■

For now it seems that $\mathcal{H}$ does not give us much in comparison with $\mathcal{P}$. On the contrary, let us go through several important properties that System $\mathcal{H}$ provably enjoys. Besides getting rid of the symbol $\lceil\cdot\rceil$ in our proof system, we have gained an easy way to reason in contexts (Section 4.2.1, Section 4.2.2), an analogue of the deduction property (Section 4.2.3), or even the so-called local completeness (Section 4.2.4).

4.2.1 *Frame reasoning*

$\mathcal{H}$ allows us to do the so-called frame reasoning. Intuitively said, frame reasoning is reasoning inside structures from the "outside". We do local calculations independently of the used data structure, which can then be "framed" into a data structure represented by a symbol context [27, p. 16]. To illustrate this, one can consider a simple example in a model of natural numbers such as[5]

$$(\mathbb{N}, +, *) \models \underline{5} * \underline{5} \to \underline{25}.$$

Suppose we want to do the same reasoning in a data structure such as a pair written as $\langle\cdot,\cdot\rangle$. For simplicity, let us assume that pairs exist in $(\mathbb{N}, +, *)$. The above property that holds "locally" can be "framed" into $\langle\cdot,\cdot\rangle$ without any additional axioms, i.e.,

$$(\mathbb{N}, +, *) \models \langle \underline{5} * \underline{5}, 2\rangle \to \langle \underline{25}, \underline{2}\rangle.$$

5 Recall that the pattern intuitively says: every element that matches $5 * 5$ also matches 25.

We do not have to axiomatize this for every data structure separately. This is because matching logic symbols (and symbol contexts) are monotonic: (FRAMING) says that

$$\rho^{\mathfrak{M}}(\psi) \subseteq \rho^{\mathfrak{M}}(\varphi) \text{ implies } \rho^{\mathfrak{M}}(C_\sigma[\psi]) \subseteq \rho^{\mathfrak{M}}(C_\sigma[\varphi]).$$

Pairs here are only used for simplicity; patterns can express much more complicated data structures such as heaps or maps [27]. The point is that we can do this kind of reasoning in System $\mathcal{H}$:

Lemma 4.2.1 (Sound Frame Reasoning [10]). Let Γ be a Σ-theory such that $\sigma \in \Sigma_n$. If $\Gamma \vdash \varphi_i \to \varphi_i'$ for all $1 \leq i \leq n$, then we also have $\Gamma \vdash \sigma(\varphi_1, \ldots, \varphi_n) \to \sigma(\varphi_1', \ldots, \varphi_n')$. ■

Notice that the rules (FRAMING), (PROPAGATION$_\bot$), (PROPAGATION$_\exists$), and finally the rule (PROPAGATION$_\vee$) are defined only for single-symbol contexts. The following lemmas say that we can generalize frame reasoning rules to nested symbol contexts.

Lemma 4.2.2 (Framing through Symbol Contexts [10]). For any nested symbol context C we have that $\Gamma \vdash \varphi \to \varphi'$ implies $\Gamma \vdash C[\varphi] \to C[\varphi']$. ■

Lemma 4.2.3 (Propagation through Symbol Contexts [10]). Let Σ be a signature and $\varphi, \varphi_1, \varphi_2 \in \text{PATTERN}_\Sigma$. For any nested symbol context C we have

(1) $\vdash C[\bot] \leftrightarrow \bot$.

(2) $\vdash C[\varphi_1 \vee \varphi_2] \leftrightarrow (C[\varphi_1] \vee C[\varphi_2])$.

(3) $\vdash C[\exists x.\, \varphi] \leftrightarrow \exists x.\, C[\varphi]$.

■

Corollary 4.2.1 ([10]). Let Γ be any Σ-theory and $\varphi, \varphi_1, \varphi_2 \in \text{PATTERN}_\Sigma$. Then the following propositions hold:

(1) $\Gamma \vdash C[\bot]$ iff $\Gamma \vdash \bot$,

(2) $\Gamma \vdash C[\varphi_1 \vee \varphi_2]$ iff $\Gamma \vdash C[\varphi_1] \vee C[\varphi_2]$,

(3) $\Gamma \vdash C[\exists x.\, \varphi]$ iff $\Gamma \vdash \exists x.\, C[\varphi]$.

■

There is one last property left to be covered when it comes to symbol contexts. In modal logic, there is a concept of the so-called *duals*. For example, the well-known diamond symbol $\Diamond$ is dual to the modal logic symbol $\Box$ because

$$\Box\varphi \equiv \neg\Diamond\neg\varphi.$$

If we have $\sigma(\varphi_1, \ldots, \varphi_n)$, then we call $\neg\sigma(\neg\varphi_1, \ldots, \neg\varphi_n)$ its dual. In this sense, the totality construct $\lfloor\varphi\rfloor \equiv \neg\lceil\neg\varphi\rceil$ is dual to $\lceil\varphi\rceil$. We can show that ML duals have much in common with modal logic duals, for example [10, p. 3]:

Lemma 4.2.4 ([10]). Let Γ be a Σ-theory and C_σ be any symbol context in Σ. Then $\Gamma \vdash \varphi$ implies $\Gamma \vdash \neg C_\sigma[\neg\varphi]$. ■

Note that $\vdash \varphi$ *implies* $\vdash \neg\Diamond(\neg\varphi)$ is a special case of Lemma 4.2.4 known as the normal modal logic generalization rule [4, p. 33].

4.2.2 *Equivalence as a congruence*

In the previous section, we saw how rules with single-symbol contexts generalize to nested symbol contexts. Here we show that $\leftrightarrow$ can be seen as a congruence. One can take any context C (not just a symbol context) and replace a pattern it contains with any other equivalent pattern while preserving validity.

Lemma 4.2.5 ([10]). Let C be any context (not just a nested symbol context). Then $\Gamma \vdash \varphi_1 \leftrightarrow \varphi_2$ implies $\Gamma \vdash C[\varphi_1] \leftrightarrow C[\varphi_2]$. ■

This answers the hanging question about equality from the end of Section 2.5. Lemma 4.2.5 is stronger than the axiom

$$\Big(\bigwedge_{1\leq i\leq n} \varphi_i = \varphi_i'\Big) \to \sigma(\varphi_1,\dots,\varphi_n) = \sigma(\varphi_1',\dots,\varphi_n')$$

because by soundness of $\mathcal{H}$ we can replace equivalent patterns in any contexts, not just symbol contexts. This also applies to equal patterns because we have $\{\lceil x\rceil\} \vdash \varphi_1 \leftrightarrow \varphi_2$ iff $\rho^{\mathfrak{M}}(\varphi_1) = \rho^{\mathfrak{M}}(\varphi_2)$ for all models $\mathfrak{M}$ with $\mathfrak{M} \models \lceil x\rceil$ and M-valuations ρ iff $\{\lceil x\rceil\} \vdash \varphi_1 = \varphi_2$.

Lemma 4.2.5 also allows us to unwrap Γ-predicates from totality because totality (and definedness) is an identity on Γ-predicates:

Proposition 4.2.3 (Totality canceling). Let ψ be a Γ-predicate. Then (1) $\Gamma_= \vdash \psi \leftrightarrow \lfloor\psi\rfloor$ and (2) $\Gamma \vdash C[\psi] \leftrightarrow C[\lfloor\psi\rfloor]$ for any context C.

Proof. (1) holds because totality is an identity on Γ-predicates (Proposition 2.5.3), i.e., $\Gamma_= \models \psi \leftrightarrow \lfloor\psi\rfloor$. (2) follows from (1) and Lemma 4.2.5. ■

4.2.3 *Deduction property*

The deduction property is one of the major properties of some Hilbert-style proof systems. Let $\mathcal{D}$ be any Hilbert-style proof system with provability denoted $\vdash_{\mathcal{D}}$ (not necessarily a system for ML). Then we say $\mathcal{D}$ has the deduction property if for every closed formula ψ,

$$\Gamma \cup \{\psi\} \vdash_{\mathcal{D}} \varphi \text{ iff } \Gamma \vdash_{\mathcal{D}} \psi \to \varphi.$$

The deduction property is very useful and tends to make logical arguments shorter. However, it is important to note that the deduction property has various variants, especially in modal logic that raises

problems for deduction [3, p. 94]. This variant we just defined is usually considered in FOL. It is natural to ask if the deduction property holds for $\mathcal{H}$. For some premises ψ, we have $\Gamma \cup \{\psi\} \vdash \varphi$ iff $\Gamma \vdash \psi \to \varphi$:

Example 4.2.1 (Explosion principle). It is easy to see that for any pattern φ we have $\Gamma \cup \{\bot\} \vdash \varphi$ iff $\Gamma \vdash \bot \to \varphi$.

- ($\Rightarrow$) Let $\Gamma \cup \{\bot\} \vdash \varphi$. Because $\vdash \bot \to \varphi$ is an instance of (PT), trivially $\Gamma \vdash \bot \to \varphi$.
- ($\Leftarrow$) Let $\Gamma \vdash \bot \to \varphi$. The proof of $\Gamma \cup \{\bot\} \vdash \varphi$ is as follows: $\bot$, $\bot \to \varphi$ (PT), apply (MP) on the first two.

Notice that for $\Gamma := \emptyset$ we get the explosion principle: for all φ we have $\{\bot\} \vdash \varphi$. ■

However, recall that symbols are interpreted in matching logic as maps to sets of model elements. The deduction property cannot hold for any sound and reasonably strong[6] proof system for matching logic such as $\mathcal{H}$. There is an easy counterexample even without (Definedness):

Proposition 4.2.4. The deduction property does not hold for $\mathcal{H}$.

Proof. By a counterexample. Here is the proof for $\{\exists x.\, \sigma(x), \lambda\} \vdash \sigma(\lambda)$:

1. λ	
2. $\lambda \to (\exists x.\, x \to \lambda)$	(PT)
3. $\exists x.\, x \to \lambda$	1., 2. (MP)
4. $\sigma(\exists x.\, x) \to \sigma(\lambda)$	3. (Framing)
5. $\sigma(\exists x.\, x) \leftrightarrow \exists x.\, \sigma(x)$	Lemma 4.2.3
6. $\exists x.\, \sigma(x)$	
7. $(\exists x.\, \sigma(x)) \to \sigma(\lambda)$	4., 5. Lemma 4.2.5
8. $\sigma(\lambda)$	6., 7. (MP)

Now we show that $\exists x.\, \sigma(x) \nvdash \lambda \to \sigma(\lambda)$. Assume for a contradiction $\exists x.\, \sigma(x) \vdash \lambda \to \sigma(\lambda)$. Soundness of $\mathcal{H}$ implies $\exists x.\, \sigma(x) \models \neg\lambda \vee \sigma(\lambda)$. However, consider the model

$$\mathfrak{M} : M := \{0, 1\}, \lambda^{\mathfrak{M}} := \{1\}, \sigma^{\mathfrak{M}}(0) := \{0, 1\}, \sigma^{\mathfrak{M}}(1) := \emptyset.$$

We have $\mathfrak{M} \models \exists x.\, \sigma(x)$ but $\mathfrak{M} \not\models \neg\lambda \vee \sigma(\lambda)$, contradiction. ■

This shows that the deduction property is semantically too strong to hold in ML. We could already see hints in Proposition 2.4.4, which says $\mathfrak{M} \models \psi \to \varphi$ iff for any M-valuation ρ we have $\rho^{\mathfrak{M}}(\psi) \subseteq \rho^{\mathfrak{M}}(\varphi)$. Such a behavior is different from how implication normally behaves

6 The deduction property holds for a complete system of propositional logic, which is a sound proof system for matching logic.

in logics such as FOL. Just consider models where ψ is not valid: we have already discussed that there is no guarantee $\psi \to \varphi$ is valid in those models.

To have any hope for an analogoue of the deduction property from FOL, we have to define a pattern that semantically captures the idea $\mathfrak{M} \models \psi$ implies $\mathfrak{M} \models \varphi$. This is only possible if we can *enforce* that the premise of the implication is either valid in $\mathfrak{M}$ or does not match any element in $\mathfrak{M}$. Observe that this can precisely be done by putting ψ into the *totality context*. We already know that $\lfloor \psi \rfloor$ is an $\mathfrak{M}$-predicate for any model $\mathfrak{M}$ with $\mathfrak{M} \models \lceil x \rceil$. What is more, we can easily show that $\lfloor \psi \rfloor \to \varphi$ does what we would expect from an implication:

The pattern $\lfloor \psi \rfloor \to \varphi$ exactly corresponds to our intuition about implication.

Proposition 4.2.5 ([27]). Let $\mathfrak{M}$ be a Σ-model such that $\mathfrak{M} \models \lceil x \rceil$. Then $\mathfrak{M} \models \lfloor \psi \rfloor \to \varphi$ iff $\mathfrak{M} \models \psi$ implies $\mathfrak{M} \models \varphi$. ■

Using the construction we have just showed in Proposition 4.2.5, we can have a weaker version of the deduction property [11, p. 4]:

Theorem 4.2.3 ("Weak" deduction property of $\mathcal{H}$ [11]). Let Γ be a Σ-theory containing (Definedness). For every closed Σ-pattern ψ we have $\Gamma \cup \{\psi\} \vdash \varphi$ iff $\Gamma \vdash \lfloor \psi \rfloor \to \varphi$. ■

Proposition 4.2.6. Let Γ be a Σ-theory and $\lceil \cdot \rceil$ be the fresh symbol in $\Gamma_=$. For every closed ψ we have $\Gamma_= \cup \{\psi\} \vdash \varphi$ iff $\Gamma_= \vdash \lfloor \psi \rfloor \to \varphi$. ■

Equality extensions are not considered by [11], but again, Proposition 4.2.6 trivially follows from their results.

The deduction property of $\mathcal{H}$ tells us something fundamental about matching logic. That is, closed Γ-predicates should exactly be those patterns for which the conventional deduction property (without premises in totality contexts) is sound. To prove this, let us first formally define this class of patterns.

Definition 4.2.2 (Γ-deduct). Let Γ be a Σ-theory and $\psi \in \text{Pattern}_\Sigma$. Then ψ is called a Γ-*deduct* if ψ is closed and for all $\varphi \in \text{Pattern}_\Sigma$ holds $\Gamma \cup \{\psi\} \vdash \varphi$ iff $\Gamma \vdash \psi \to \varphi$. ■

For example, $\bot$ is a Γ-deduct for any theory Γ (Example 4.2.1). In theories Γ containing (Definedness), closed Γ-predicates exactly correspond to Γ-deducts (and vice versa):

Theorem 4.2.4. Let Γ be a Σ-theory containing (Definedness) and $\psi \in \text{Pattern}_\Sigma$ some *closed* pattern. Then ψ is a Γ-predicate iff ψ is a Γ-deduct.

Proof. Let $\varphi \in \text{Pattern}_\Sigma$.

($\Rightarrow$) Let ψ be a Γ-predicate. We want to prove ψ is a Γ-deduct. Let $\Gamma \cup \{\psi\} \vdash \varphi$. By weak deduction property we have $\Gamma \vdash \lfloor \psi \rfloor \to \varphi$. Because ψ is a Γ-predicate, by totality canceling (Proposition 4.2.3) $\Gamma \vdash \psi \to \varphi$. The other implication $\Gamma \vdash \psi \to \varphi$ implies $\Gamma \cup \{\psi\} \vdash \varphi$ is trivial by (MP).

($\Leftarrow$) By contraposition. If ψ is not a Γ-predicate, there is some model $\mathfrak{M} \models \Gamma$ and M-valuation ρ such that $\emptyset \subset \rho^{\mathfrak{M}}(\psi) \subset M$. We show that

ψ is not a Γ-deduct by counterexample. Obviously $\Gamma \cup \{\psi\} \vdash \lfloor\psi\rfloor$ and $\mathfrak{M}$ is a witness to $\Gamma \nvdash \psi \to \lfloor\psi\rfloor$ by completeness of $\mathcal{H}$ w.r.t. Γ (Proposition 4.2.2). ■

Corollary 4.2.2. Let Γ be a Σ-theory and $\psi \in \textsc{Pattern}_{\Sigma_=}$ some *closed* pattern. Then ψ is a $\Gamma_=$-predicate iff ψ is a $\Gamma_=$-deduct.

Proof. $\Gamma_=$ contains a definedness axiom with some fresh symbol. Repeat the proof of Theorem 4.2.4 with this symbol. ■

In Chapter 5 we shall prove that completeness of $\mathcal{H}$ implies that Γ-predicates are Γ-deducts even if Γ does not contain (Definedness).

4.2.4 *Local completeness*

Besides the fact that $\mathcal{H}$ is complete w.r.t. theories containing the axiom (Definedness), $\mathcal{H}$ is complete for empty theories (Theorem 4.2.5). The proof draws inspiration from [5] and is rather technical, combining techniques from both hybrid modal logic and first-order logic [10, p. 4]. We provide the result here only for reference.

Theorem 4.2.5 (Local completeness [10]). Let φ be a pattern. Then $\emptyset \models \varphi$ implies $\emptyset \vdash \varphi$. ■

Notice that completeness of $\mathcal{H}$ wrt the empty theory holds without any further conditions. Especially we do not need the axiom (Definedness), of course. If we were considering FOL instead of ML, local completeness would be enough to prove completeness! In FOL, local completeness is as strong as (global) completeness:

Theorem 4.2.6. Let $\mathcal{D}$ be a sound Hilbert-style proof system for FOL with modus ponens. If for all valid FOL formulas φ we have $\emptyset \vdash_{\mathcal{D}} \varphi$, then $\mathcal{D}$ is a complete proof system for FOL ($\Phi \models \varphi$ implies $\Phi \vdash \varphi$).

Proof. Let $\emptyset \vdash_{\mathcal{D}} \varphi$ for all formulas φ such that $\emptyset \models_{\mathsf{FOL}} \varphi$. We want to show $\Phi \models \varphi$ implies $\Phi \vdash \varphi$ for every FOL theory Φ. It is enough to show that every consistent theory Φ has a model (Henkin's reduction [20]), i.e., $\Phi \nvdash_{\mathcal{D}} \bot_{\mathsf{FOL}}$ implies $\Phi \not\models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$ for some FOL contradiction $\bot_{\mathsf{FOL}}$.

Let Φ be an FOL theory such that $\Phi \nvdash_{\mathcal{D}} \bot_{\mathsf{FOL}}$. The compactness property of FOL yields $\Phi \not\models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$ iff $\Phi_{\mathsf{fin}} \not\models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$ for every finite subset $\Phi_{\mathsf{fin}} \subseteq \Phi$. We can w.l.o.g. assume that Φ contains only closed formulas. Thus by FOL semantics this is iff

$$\emptyset \not\models_{\mathsf{FOL}} \bigwedge \Phi_{\mathsf{fin}} \to \bot_{\mathsf{FOL}} \text{ for every finite subset } \Phi_{\mathsf{fin}} \subseteq \Phi.$$

Let $\Phi_{\mathsf{fin}} \subseteq \Phi$ be any finite subset of Φ. By definition of $\vdash_{\mathcal{D}}$ we get $\Phi_{\mathsf{fin}} \nvdash_{\mathcal{D}} \bot_{\mathsf{FOL}}$. But this means $\emptyset \nvdash_{\mathcal{D}} \bigwedge \Phi_{\mathsf{fin}} \to \bot_{\mathsf{FOL}}$ (because assuming $\emptyset \vdash_{\mathcal{D}} \bigwedge \Phi_{\mathsf{fin}} \to \bot_{\mathsf{FOL}}$ leads to an obvious contradiction). Finally local completeness of $\mathcal{D}$ yields $\not\models_{\mathsf{FOL}} \bigwedge \Phi_{\mathsf{fin}} \to \bot_{\mathsf{FOL}}$. ■

Theorem 4.2.6 shows that it is enough to show local completeness of an FOL proof system to show that this system is (globally) complete. In fact, Godel's original proof of completeness is proving local completeness of a FOL proof system [16]. In ML, however, $\Gamma \cup \{\psi\} \models \bot$ does generally not imply $\Gamma \models \psi \to \bot$. Recall that this property holds only for closed Γ-predicates (Corollary 2.4.2). Even if we chose ψ to be a Γ-predicate, we cannot repeat the same trick as in Theorem 4.2.6 because removing assumptions from a finite theory adds models, i.e., cancels some predicates of the original theory.

5

IS SYSTEM H COMPLETE?

We have learned in Section 4.2 that $\mathcal{H}$ is complete w.r.t. theories containing (Definedness), or more generally, w.r.t. equality extensions. However, $\mathcal{H}$ is sound for any ML theory. In this section we investigate if System $\mathcal{H}$ is complete w.r.t. any given theory, i.e., if

$$\Gamma \models \varphi \text{ implies } \Gamma \vdash \varphi$$

holds for all Σ-theories Γ, even if Γ does not contain (Definedness).

Recall that ML is embedabble in predicate logic with equality (Section 3.1), which has a complete proof system. That is why it was conjectured in [27, p. 57] that ML should also admit a complete proof system, which does not rely on any predefined symbols such as $\lceil \cdot \rceil$. System $\mathcal{H}$ does not contain any predefined symbols and we have seen indications that $\mathcal{H}$ is quite a strong proof system. Among other things, $\mathcal{H}$ is a locally complete proof system for ML (Theorem 4.2.5), i.e., it is complete for the empty theory $\Gamma = \emptyset$. On the other hand, we need predicate logic *with equality* to contain ML. Local deduction $\vdash \varphi$ seems "weaker" than global deduction $\Gamma \vdash \varphi$. This has been explained by our discussions about local completeness in Section 4.2.4 and the deduction property in Section 4.2.3, where we have seen that

$$\{\psi\} \vdash \varphi \text{ does not imply } \emptyset \vdash \psi \to \varphi.$$

We shall see that the deduction property plays a central role in the question whether $\mathcal{H}$ is complete.

The structure of this chapter is as follows. We try to follow Henkin's method of proving completeness for FOL [20] to see exactly where it fails for ML: the deduction property. This will lead us to an alternative characterization of completeness, which is the main result of this thesis. We notice we can formalize the notion that (Definedness) "makes" $\mathcal{H}$ complete. It turns out $\mathcal{H}$ is complete if and only if every equality extension is a *conservative extension*. It is difficult to show that an extension is conservative without a complete proof system and the deduction property. However, this result will allow us to reduce the problem of completeness to finite theories and prove some *instances* of completeness. By an instance of completeness we mean that $\mathcal{H}$ is complete w.r.t. some given class of theories:

Definition 5.0.1 ($\mathcal{H}$-complete theory). Let Γ be a Σ-theory. We say that $\mathcal{H}$ is complete w.r.t. Γ (or simply that Γ is *$\mathcal{H}$-complete*) iff for every $\varphi \in \text{Pattern}_\Sigma$ we have that $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$. ■

For example, all theories containing (Definedness) are $\mathcal{H}$-complete in this sense. If we find that $\mathcal{H}$ is not complete because of something fundamental about ML (and not just because of some missing rules), we would at least like to *characterize* $\mathcal{H}$-complete theories and prove results about them. That means, we would like an if-and-only-if condition that tells us whether a theory is $\mathcal{H}$-complete.

Let us recap how we proceed in the following list.

(1) We show where Henkin's method fails when applied to ML. Then we find an alternative characterization of $\mathcal{H}$-complete theories: we reduce the completeness problem of $\mathcal{H}$ to proving that every equality extension is a *conservative extension* (Section 5.1).

(2) We show that we can reduce completeness of $\mathcal{H}$ to completeness of $\mathcal{H}$ w.r.t. *finite* theories (Section 5.2).

(3) We prove some instances of completeness. In particular, we show that $\mathcal{H}$ is complete w.r.t. the fragment of ML without symbols (Section 5.3). If a theory does not contain any symbols, we will show that this theory is $\mathcal{H}$-complete.

(4) We study the notion of $\mathcal{H}$-consistency. We show that $\mathcal{H}$-consistent theories have similar properties to consistent theories in FOL. In particular, we use this to prove the well-known *compactness* property for ML (Section 5.4).

(5) We study the notion of *negation-complete* theories in ML (Section 5.5).

(6) We conclude the chapter with open leads to proving other instances of completeness.

5.1 AN IF-AND-ONLY-IF CONDITION FOR COMPLETENESS

Henkin showed that instead of solving completeness of an FOL proof system $\mathcal{D}$, we can show that every $\mathcal{D}$-consistent theory has a model [20]. This allowed to construct the so-called *canonical model* for a given (consistent) theory, which yielded a technique to prove completeness in different variants for many logics [5, 12, 13]. Since ML is a variant of FOL (Chapter 3), why not try Henkin's method?

Theorem 5.1.1 (Henkin's reduction [20]). Let $\mathcal{D}$ be an FOL sound Hilbert-style proof system with the deduction property that has at least modus ponens and such that $\vdash_{\mathcal{D}} (\neg\varphi \to \bot_{\mathsf{FOL}}) \to \varphi$ for some FOL contradiction $\bot_{\mathsf{FOL}}$. Then the following two statements are equivalent:

(1) $\mathcal{D}$ is a complete proof system ($\Phi \models \varphi$ implies $\Phi \vdash \varphi$).

(2) Every $\mathcal{D}$-consistent theory has a model.

Proof.

($\Rightarrow$) Let Φ be a $\mathcal{D}$-consistent theory, i.e., $\Phi \nvdash_{\mathcal{D}} \bot_{\mathsf{FOL}}$. Then by completeness of $\mathcal{D}$ we get $\Phi \not\models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$. By definition of $\models_{\mathsf{FOL}}$ this means that there exists a model $\mathfrak{A}$ of Φ such that $\mathfrak{A} \not\models_{\mathsf{FOL}} \bot_{\mathsf{FOL}}$, i.e., Φ is satisfiable.

($\Leftarrow$) We want to prove that $\mathcal{D}$ is complete, i.e., $\Phi \nvdash_{\mathcal{D}} \varphi$ implies $\Phi \not\models_{\mathsf{FOL}} \varphi$. Let $\Phi \nvdash_{\mathcal{D}} \varphi$.

- First we show that $\Phi \cup \{\neg\varphi\}$ is satisfiable. Assume for a contradiction that $\Phi \cup \{\neg\varphi\}$ is unsatisfiable. (2) yields that every $\mathcal{D}$-consistent theory is satisfiable, which is the same as saying every unsatisfiable theory is $\mathcal{D}$-inconsistent. Thus (2) yields $\Phi \cup \{\neg\varphi\} \vdash_{\mathcal{D}} \bot_{\mathsf{FOL}}$. The deduction property yields

  $$\Phi \vdash_{\mathcal{D}} \neg\varphi \to \bot_{\mathsf{FOL}}.$$

  But now we can use $\vdash_{\mathcal{D}} (\neg\varphi \to \bot_{\mathsf{FOL}}) \to \varphi$ and modus ponens to get $\Phi \vdash_{\mathcal{D}} \varphi$, contradiction.

- Because $\Phi \cup \{\neg\varphi\}$ is satisfiable, there exists a model $\mathfrak{A}$ of $\Phi \cup \{\neg\varphi\}$, i.e., $\mathfrak{A} \models_{\mathsf{FOL}} \Phi \cup \{\neg\varphi\}$. By definition of $\models_{\mathsf{FOL}}$ we immediately get that $\mathfrak{A} \models_{\mathsf{FOL}} \Phi$ and $\mathfrak{A} \models \neg\varphi$. However, $\mathfrak{A} \models_{\mathsf{FOL}} \neg\varphi$ implies $\mathfrak{A} \not\models_{\mathsf{FOL}} \varphi$. Thus $\mathfrak{A} \models \Phi$ and $\mathfrak{A} \not\models \varphi$, i.e., $\Phi \not\models_{\mathsf{FOL}} \varphi$.

■

Besides a few technicalities, the main idea in the proof of Theorem 5.1.1 is applying the deduction property. Theorem 5.1.1 can be seen as a *reduction* because it reduces the problem of completeness to the problem of showing that every consistent theory has a model. If we want to use this for ML, we get immediately stuck because we do not have the deduction property (Section 4.2.3). Can we too "reduce" the completeness problem of $\mathcal{H}$ to something more tractable?

In Chapter 4 we have learned that $\mathcal{H}$ is complete w.r.t. theories containing (Definedness). Intuitively said, (Definedness) "makes" each theory $\mathcal{H}$-complete. Is it not a way how to characterize $\mathcal{H}$-complete theories? We do not need to restrict ourselves to (Definedness): all equality extensions are $\mathcal{H}$-complete (Proposition 4.2.2). Equality extensions are very simple because we only add a single axiom with a fresh symbol. We have learned in Proposition 2.2.2 that models restricted to a smaller signature evaluate patterns in the smaller signature to the same set as the original models do. If we have a $\Sigma \cup \{\lceil\cdot\rceil\}$-model $\mathfrak{M}$ with $\mathfrak{M} \models \lceil x \rceil$, then for every M-valuation ρ holds $\rho^{\mathfrak{M}}(\varphi) = \rho^{\mathfrak{M}|_\Sigma}(\varphi)$ for $\varphi \in \textsc{Pattern}_\Sigma$. This concept should sound familiar. An equality extension is indeed a model-theoretic *conservative extension*:

Definition 5.1.1. Let Γ be a Σ-theory and Γ^+ be a Σ^+-theory such that $\Sigma \subseteq \Sigma^+$. We say that Γ^+ is a *model-theoretic conservative extension* of Γ if for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma^+ \models \varphi$ iff $\Gamma \models \varphi$. ■

Lemma 5.1.1 (Extension by definedness). Let Γ be a Σ-theory. Then $\Gamma_=$ is a model-theoretic conservative extension of Γ, i.e., for every Σ-pattern φ we have $\Gamma_= \models \varphi$ iff $\Gamma \models \varphi$.

Proof. This is immediate from Proposition 2.2.2 but we will show a full proof. Let $\lceil\cdot\rceil$ be w.l.o.g. the fresh definedness symbol in $\Gamma_=$.

($\Rightarrow$) By contraposition. Let $\Gamma \not\models \varphi$. By definition of $\models$ we have a Σ-model $\mathfrak{M} = (M, I)$ of Γ such that $\mathfrak{M} \not\models \varphi$. Consider the model $\mathfrak{M}^+ = (M, I \cup \{\lceil\cdot\rceil^{\mathfrak{M}^+}\})$ where $\lceil\cdot\rceil^{\mathfrak{M}^+}(m) = M$ for all $m \in M$. But then obviously $\mathfrak{M}^+ \models \Gamma \cup \{\lceil x\rceil\}$ and $\mathfrak{M}^+ \not\models \varphi$ because $\lceil\cdot\rceil \notin \Sigma$. This means $\Gamma_= \not\models \varphi$.

($\Leftarrow$) By contraposition. Let $\Gamma_= \not\models \varphi$. By definition of $\models$ there exists a model $\mathfrak{M}$ of $\Gamma_=$ such that $\mathfrak{M} \not\models \varphi$. Consider that also $\mathfrak{M}|_\Sigma \models \Gamma$ because $\Gamma \subset \Gamma_=$ and Γ is a Σ-theory. Because $\varphi \in \textsc{Pattern}_\Sigma$, we also have $\mathfrak{M}|_\Sigma \not\models \varphi$. But then $\mathfrak{M}|_\Sigma \models \Gamma$ and $\mathfrak{M}|_\Sigma \not\models \varphi$, i.e., $\Gamma \not\models \varphi$. ■

We can exploit Lemma 5.1.1 to reduce completeness of $\mathcal{H}$ to proving that every equality extension is a *proof-theoretic* conservative extension in $\mathcal{H}$. There is indeed a good intuition behind because $\mathcal{H}$ is complete w.r.t. equality extensions (Proposition 4.2.2). We start with the definition of proof-theoretic extensions.

Definition 5.1.2 (Extension). Let Γ be a Σ-theory and Γ' be a Σ'-theory. If $\Sigma \subseteq \Sigma'$ and for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma \vdash \varphi$ implies $\Gamma' \vdash \varphi$, then Γ' is called a (proof-theoretic) *extension* of Γ. ■

Note that $\Gamma_=$ is a proof-theoretic extension of Γ in the sense of Definition 5.1.2 trivially because $\Gamma \subset \Gamma_=$. Conservative extensions also satisfy the other direction $\Gamma' \vdash \varphi$ implies $\Gamma \vdash \varphi$ for $\varphi \in \textsc{Pattern}_\Sigma$, i.e., the original theory can prove everything in the original signature that the extension can:

Definition 5.1.3 (Conservative extension). Let Γ be a Σ-theory and Γ' be a Σ'-theory. If Γ' is an extension of Γ and for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma' \vdash \varphi$ implies $\Gamma \vdash \varphi$, then Γ' is called a (proof-theoretic) *conservative extension* of Γ. ■

We will always drop the adjective proof-theoretic when we refer to proof-theoretic extensions or proof-theoretic conservative extensions. Because we know that $\mathcal{H}$ is complete w.r.t. equality extensions (Proposition 4.2.2), it is natural to ask how equality extensions relate to completeness of $\mathcal{H}$. We know that $\Gamma \models \varphi$ iff $\Gamma_= \models \varphi$ for φ in the signature of Γ (Lemma 5.1.1). But we also know that $\Gamma_= \models \varphi$ iff $\Gamma_= \vdash \varphi$! The problem whether Γ is $\mathcal{H}$-complete can be reduced to showing that $\Gamma_=$ is a conservative extension of Γ. If we prove this for every Γ, then $\mathcal{H}$ is complete:

Theorem 5.1.2. Let Γ be a Σ-theory. The following two statements are equivalent:

(1) For every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$,
(Γ is $\mathcal{H}$-complete)

(2) For every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma_= \vdash \varphi$ implies $\Gamma \vdash \varphi$.
($\Gamma_=$ is a conserative extension of Γ)

Proof.
($\Rightarrow$) Let $\varphi \in \textsc{Pattern}_\Sigma$. We show the contraposition of (2). If $\Gamma \nvdash \varphi$, by (1) we have $\Gamma \not\models \varphi$. By the extension by definedness lemma (Lemma 5.1.1) we have $\Gamma_= \not\models \varphi$. By correctness of $\mathcal{H}$ we finally get $\Gamma_= \nvdash \varphi$.
($\Leftarrow$) Let $\varphi \in \textsc{Pattern}_\Sigma$. We show the contraposition of (1). If $\Gamma \nvdash \varphi$, by (2) we have $\Gamma_= \nvdash \varphi$. By completeness of $\mathcal{H}$ w.r.t. $\Gamma_=$, we have $\Gamma_= \not\models \varphi$. By extension by definedness lemma (Lemma 5.1.1) we finally get $\Gamma \not\models \varphi$. ■

Corollary 5.1.1 (Characterization of completeness). The following two statements are equivalent:

(1) $\mathcal{H}$ is complete, i.e., every Σ-theory Γ is $\mathcal{H}$-complete.

(2) For every Σ-theory Γ we have that $\Gamma_=$ is a conservative extension of Γ.

Here is why equality extensions are actually useful.

■

We can delve even deeper. The question of completeness of $\mathcal{H}$ is really an instance of the problem whether *extensions by definition*[1] [13, p. 126] are conservative extensions in $\mathcal{H}$: in FOL, an extension by definition of a fresh n-ary predicate symbol P is extending a given theory with a so-called definition φ of P

$$\forall x_1 \ldots \forall x_n . P(x_1, \ldots, x_n) \leftrightarrow \varphi(x_1, \ldots, x_n)$$

where φ is in the original signature. Extensions by definition are trivially conservative in FOL because FOL has a complete proof system.

Notice that equality extensions are extensions by definition in this sense because $\lceil x \rceil$ really says $\forall x. \lceil x \rceil \leftrightarrow \top$. In ML, it is possible to show that extensions by definition mean almost the same thing as in FOL and thus they should be conservative; semantically, this makes sense as we have already seen. If $\mathcal{H}$ is complete, extensions by definition in ML are conservative. We are in a situation where we would like to show the other direction; this is difficult.

However, all is not lost. Notice that Theorem 5.1.2 allows us to prove completeness of $\mathcal{H}$ on a case-by-case basis. Our if-and-only-if condition for $\mathcal{H}$-complete theories gives a straightforward manner of solving completeness of $\mathcal{H}$ w.r.t. some theories Γ. For example, $\mathcal{H}$ is complete w.r.t. predicate patterns ($\emptyset$-predicates). The idea is the analogous to

1 Mind the difference between a definition and a definedness axiom.

the one used in FOL for reducing completeness to local completeness (Theorem 4.2.6), i.e., we remove patterns from our assumptions "to the front":

Theorem 5.1.3. Let $\psi \in \textsc{Pattern}_\Sigma$ be a $\emptyset$-predicate. Then $\{\psi\}$ is $\mathcal{H}$-complete.

Proof. We will prove an equivalent statement that $\{\psi\}_=$ is a conservative extension of $\{\psi\}$ (Theorem 5.1.2), i.e., for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\{\psi\}_= \vdash \varphi$ implies $\{\psi\} \vdash \varphi$.

Let $\{\psi\}_= \vdash \varphi$, which is the same as $\{\lceil x \rceil\} \cup \{\psi\} \vdash \varphi$. Recall that ψ is an $\emptyset$-predicate, thus it is also a $\{\lceil x \rceil\}$-predicate by Lemma 5.1.1. But then ψ is a $\{\lceil x \rceil\}$-deduct by Theorem 4.2.4, thus $\{\lceil x \rceil\} \vdash \psi \to \varphi$. By soundness we get $\{\lceil x \rceil\} \models \psi \to \varphi$. But then extension by definedness lemma (Lemma 5.1.1) yields $\models \psi \to \varphi$ because obviously $\psi \to \varphi \in \textsc{Pattern}_\Sigma$. By local completeness of $\mathcal{H}$ we get $\vdash \psi \to \varphi$. The proof of $\{\psi\} \vdash \varphi$ is introducing ψ, applying $\vdash \psi \to \varphi$ and applying (MP) on ψ and $\psi \to \varphi$. ■

Notice that Theorem 5.1.3 solves the question whether $\{\forall x.\, x\}$ is $\mathcal{H}$-complete mentioned in Section 4.1 because $\chi \equiv \forall x.\, x$ is a predicate pattern! The pattern χ evaluates to $\bigcap_{m \in M}\{m\}$ in every model $\mathfrak{M}$. Therefore in single-element models the pattern χ is trivially valid, in other models χ matches no element because the intersection is empty. This intuition can be extended further to all closed patterns without symbols, which is the main idea behind solving completeness of $\mathcal{H}$ w.r.t. the fragment without symbols (Section 5.3).

An attentive reader might have also noticed that our if-and-only-if condition for completness (Theorem 5.1.2) not only leads to a potential proof of completeness of $\mathcal{H}$. It also means that $\mathcal{H}$-complete theories admit a stronger version of the weak deduction property:

Proposition 5.1.1. Let Γ be an $\mathcal{H}$-complete Σ-theory. Then for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma \cup \{\psi\} \vdash \varphi$ iff $\Gamma_= \vdash \lfloor \psi \rfloor \to \varphi$.

Proof. Γ is $\mathcal{H}$-complete implies that $\Gamma_=$ is a conservative extension of Γ. But then for every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma \cup \{\psi\} \vdash \varphi$ iff $\Gamma_= \cup \{\psi\} \vdash \varphi$ iff $\Gamma_= \vdash \lfloor \psi \rfloor \to \varphi$, where the last equivalence is the weak deduction property. ■

This leads us back to the discussion in Section 4.2.3 about Γ-deducts, i.e., patterns for which the conventional deduction property holds. The deduction property and our if-and-only-if condition for the completeness of $\mathcal{H}$ using conservative extensions (Theorem 5.1.2) implies that Γ-predicates are Γ-deducts for any $\mathcal{H}$-complete theory Γ.

Theorem 5.1.4. Let Γ be an $\mathcal{H}$-complete theory. Then every closed Γ-predicate is a Γ-deduct.

Proof. Let Γ be a Σ-theory and $\psi \in \textsc{Pattern}_\Sigma$ any closed Γ-predicate.

- We want to show $\Gamma \cup \{\psi\} \vdash \varphi$ implies $\Gamma \vdash \psi \to \varphi$. Let $\Gamma \cup \{\psi\} \vdash \varphi$ for some $\varphi \in \textsc{Pattern}_\Sigma$. By Proposition 5.1.1 we have $\Gamma_= \vdash \lfloor\psi\rfloor \to \varphi$. Observe that ψ must also be a $\Gamma_=$-predicate (Proposition 2.4.3) and thus totality canceling (Proposition 4.2.3) yields $\Gamma_= \vdash \psi \to \varphi$. But Theorem 5.1.2 says that $\Gamma_=$ is a conservative extension of Γ. Therefore $\Gamma \vdash \psi \to \varphi$ because $\psi \to \varphi \in \textsc{Pattern}_\Sigma$.
- We want to show $\Gamma \vdash \psi \to \varphi$ implies $\Gamma \cup \{\psi\} \vdash \varphi$. Let $\Gamma \vdash \psi \to \varphi$. Then trivially $\Gamma \cup \{\psi\} \vdash \psi \to \varphi$. The proof of $\Gamma \cup \{\psi\} \vdash \varphi$ is ψ, applying $\vdash \psi \to \varphi$, and applying (MP) on ψ and $\psi \to \varphi$.

■

Interestingly enough, the question whether deducts are predicates even in $\mathcal{H}$-complete theories is non-trivial; we do not know anything about contained symbols. We leave the general case as a conjecture.

Conjecture 5.1.1. Let Γ be an $\mathcal{H}$-complete theory. Then every Γ-deduct is a Γ-predicate.

5.2 REDUCTION TO FINITE THEORIES

Our if-and-only-if condition for completeness of $\mathcal{H}$ has an immediate consequence. The condition reduces completeness, a problem of both semantic and syntactic nature, to a strictly syntactic problem. Provability means there is a *finite* Hilbert-style proof that uses a *finite* number of axioms. If we prove the condition given by Theorem 5.1.2 for all finite theories, we will thus prove the condition for all theories. This is formalized in the following theorem.

Theorem 5.2.1 (Finite theories). The following two statements are equivalent.

(1) For every *finite* Σ-theory Γ and every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma_= \vdash \varphi$ implies $\Gamma \vdash \varphi$

(every finite equality extension is a conservative extension).

(2) For every Σ-theory Γ and every $\varphi \in \textsc{Pattern}_\Sigma$ we have $\Gamma_= \vdash \varphi$ implies $\Gamma \vdash \varphi$

(every equality extension is a conservative extension).

Proof.

($\Rightarrow$) Let $\Gamma_= \vdash \varphi$ and w.l.o.g assume $\Gamma_= \setminus \Gamma = \{\lceil x \rceil\}$ where $\lceil\cdot\rceil \notin \Sigma$. By definition of $\vdash$ there is some finite theory $\Gamma' \subseteq \Gamma_=$ such that $\Gamma' \vdash \varphi$.

Note that Γ' is not necessarily a Σ-theory because Γ' can contain $\lceil x \rceil$. That is why we instead consider the Σ-theory $\Gamma'' = \Gamma' \setminus \{\lceil x \rceil\}$. By our assumption from Definition 2.6.1 about the fixed fresh symbol for Σ we have $\Gamma' \subseteq \Gamma''_= \subseteq \Gamma_=$. But then obviously $\Gamma''_= \vdash \varphi$. Because Γ'' is

finite, we can apply (1) to get $\Gamma'' \vdash \varphi$. But then trivially $\Gamma \vdash \varphi$ because $\Gamma'' \subseteq \Gamma$.

($\Leftarrow$) Trivial. ■

Note that in the proof of Theorem 5.2.1, we have to consider Γ'' instead of Γ' because Γ' might contain the fresh definedness axiom, i.e., we might have $\Gamma' \not\subseteq \Gamma$. That is why (1) applied on $\Gamma'_=$ to get Γ' would not help us to prove the conclusion $\Gamma \vdash \varphi$. Because our if-and-only-if characterization of completeness, Theorem 5.2.1 means that we only have to prove completeness of $\mathcal{H}$ for finite theories:

Corollary 5.2.1. The following two statements are equivalent.

- Every finite Σ-theory is $\mathcal{H}$-complete.
- Every Σ-theory is $\mathcal{H}$-complete.

■

5.3 THEORIES WITHOUT SYMBOLS ARE H-COMPLETE

Now we are ready to show that $\mathcal{H}$ is complete for Σ-theories Γ with $\Sigma = \emptyset$. We can show this by proving something stronger. The fragment of ML without symbols is very weak. In fact, we will show that closed patterns without symbols are predicate patterns ($\emptyset$-predicate)! This will immediately yield completeness for matching logic without symbols because we have learned that

(1) $\{\psi\}$ is $\mathcal{H}$-complete if ψ is an $\emptyset$-predicate (Theorem 5.1.3), and

(2) every finite $\emptyset$-theory is $\mathcal{H}$-complete implies every $\emptyset$-theory is $\mathcal{H}$-complete (Theorem 5.2.1).

The main insight is the following: if we do not have symbols, we do not have a way how to *distinguish* between individual model elements. That is why we introduce some theory on *permutations* over model elements. Let $\pi : M \to M$ be a permutation (bijection) of model elements from $\mathfrak{M}$. Notice that for any M-valuation ρ, the composition $\pi \circ \rho$ is again an M-valuation. For any $X \subseteq M$, we can naturally extend permutations of elements to *sets* of model elements as follows:

$$\pi(X) = \{\pi(x) \mid x \in X\}.$$

When we apply this permutations to pattern valuations, we can show that they have several interesting properties:

Lemma 5.3.1. Let $\mathfrak{M}$ be a Σ-model, $\pi : M \to M$ any permutation and ρ any M-valuation. Then the following properties hold:

(1) $\pi(\rho^{\mathfrak{M}}(\varphi_1) \cap \rho^{\mathfrak{M}}(\varphi_2)) = \pi(\rho^{\mathfrak{M}}(\varphi_1)) \cap \pi(\rho^{\mathfrak{M}}(\varphi_2))$

(2) $\pi(M \setminus \rho^{\mathfrak{M}}(\varphi)) = M \setminus \pi(\rho^{\mathfrak{M}}(\varphi))$.

(3) $\pi(\bigcup_{m\in M}\rho[m/x]^{\mathfrak{M}}(\varphi)) = \bigcup_{m\in M}\pi(\rho[m/x]^{\mathfrak{M}}(\varphi))$.

(4) $\pi\circ\rho[m/x] = (\pi\circ\rho)[\pi(m)/x]$.

Proof. Let us prove each property on its own.

(1) We are proving the following equality

$$\{\pi(m) \mid m \in \rho^{\mathfrak{M}}(\varphi_1)\cap\rho^{\mathfrak{M}}(\varphi_2)\} = \{\pi(m) \mid m \in \rho^{\mathfrak{M}}(\varphi_1)\}\cap\{\pi(m) \mid m \in \rho^{\mathfrak{M}}(\varphi_2)\}.$$

($\subseteq$) is trivial, for ($\supseteq$) consider m that is in both sets. We know then there exists $a \in \rho^{\mathfrak{M}}(\varphi_1), b \in \rho^{\mathfrak{M}}(\varphi_2)$ such that $\pi(a) = m$ and $\pi(b) = m$. By injectivity of π we get $a = b \in \rho^{\mathfrak{M}}(\varphi_1)\cap\rho^{\mathfrak{M}}(\varphi_2)$.

(2) We prove both subsumptions.

- ($\subseteq$) Let $m \in M \setminus \pi(\rho^{\mathfrak{M}}(\varphi))$. Because π is surjective, there exists $a \in M$ such that $\pi(a) = m$. Assume for a contradiction that $a \notin M \setminus \rho^{\mathfrak{M}}(\varphi)$. Then $a \in \rho^{\mathfrak{M}}(\varphi)$, which means $m \in \pi(\rho^{\mathfrak{M}}(\varphi))$ and thus $m \notin M \setminus \pi(\rho^{\mathfrak{M}}(\varphi))$, contradiction.
- ($\supseteq$) By contraposition. Let $m \notin M \setminus \pi(\rho^{\mathfrak{M}}(\varphi))$. Thus $m \in \pi(\rho^{\mathfrak{M}}(\varphi))$. Assume for a contradiction that $m \in \pi(M \setminus \rho^{\mathfrak{M}}(\varphi))$. Thus $m \in \pi(\rho^{\mathfrak{M}}(\varphi)) \cap \pi(M \setminus \rho^{\mathfrak{M}}(\varphi))$, which by (1) means $m \in \pi(\rho^{\mathfrak{M}}(\varphi) \cap (M \setminus \rho^{\mathfrak{M}}(\varphi))) = \pi(\emptyset) = \emptyset$, contradiction.

(3) $\pi(b) \in \pi(\bigcup\limits_{m\in M}\rho[m/x]^{\mathfrak{M}}(\varphi))$ iff $b \in \bigcup\limits_{m\in M}\rho[m/x]^{\mathfrak{M}}(\varphi)$
iff $\pi(b) \in \bigcup\limits_{m\in M}\pi(\rho[m/x]^{\mathfrak{M}}(\varphi))$.

(4) We consider two cases:

- $y = x$: $(\pi\circ\rho[m/x])(y) = \pi(m) = (\pi\circ\rho)[\pi(m)/x](y)$
- $y \neq x$: $(\pi\circ\rho[m/x])(y) = \pi(\rho(y)) = (\pi\circ\rho)(y) = (\pi\circ\rho)[\pi(m)/x](y)$.

■

Recall that given an M-valuation ρ and a permutation π, $\pi\circ\rho$ is also an M-valuation. When we do not have formal symbols in the signature, the following lemma intuitively says that $(\pi\circ\rho)^{\mathfrak{M}}(\varphi)$ matches the same elements as π applied to the set $\rho^{\mathfrak{M}}(\varphi)$.

Lemma 5.3.2. Let $\mathfrak{M}$ be a $\emptyset$-model. Then for every $\emptyset$-pattern φ, every M-valuation ρ and every permutation $\pi : M \to M$ we have

$$\pi(\rho^{\mathfrak{M}}(\varphi)) = (\pi\circ\rho)^{\mathfrak{M}}(\varphi).$$

Proof. By structural induction on φ.

- $\varphi \equiv x$. By definition

$$\pi(\rho^{\mathfrak{M}}(x)) = \{\pi(a) \mid a \in \rho^{\mathfrak{M}}(x)\} = \{(\pi \circ \rho)(x)\} = (\pi \circ \rho)^{\mathfrak{M}}(x).$$

- $\varphi \equiv \neg\varphi_1$. Consider the following.

$$\begin{aligned}\pi(\rho^{\mathfrak{M}}(\varphi)) = \pi(M \setminus \rho^{\mathfrak{M}}(\varphi_1)) &\overset{1}{=} M \setminus \pi(\rho^{\mathfrak{M}}(\varphi_1)) \\ &\overset{\text{IH}}{=} M \setminus (\pi \circ \rho)^{\mathfrak{M}}(\varphi_1) \\ &= (\pi \circ \rho)^{\mathfrak{M}}(\varphi)\end{aligned}$$

  Equality (1) works is given by Lemma 5.3.1.

- $\varphi \equiv \varphi_1 \wedge \varphi_2$. Consider the following.

$$\begin{aligned}\pi(\rho^{\mathfrak{M}}(\varphi)) &= \pi(\rho^{\mathfrak{M}}(\varphi_1) \cap \rho^{\mathfrak{M}}(\varphi_2)) \\ &\overset{1}{=} \pi(\rho^{\mathfrak{M}}(\varphi_1)) \cap \pi(\rho^{\mathfrak{M}}(\varphi_2)) \\ &\overset{\text{IH}}{=} (\pi \circ \rho)^{\mathfrak{M}}(\varphi_1) \cap (\pi \circ \rho)^{\mathfrak{M}}(\varphi_2) \\ &\overset{2}{=} (\pi \circ \rho)^{\mathfrak{M}}(\varphi_1 \wedge \varphi_2)\end{aligned}$$

  (1) is given by Lemma 5.3.1. (2) is by definition of pattern interpretation because $\pi \circ \rho$ is an M-valuation.

- $\varphi \equiv \exists x.\, \varphi$.

$$\begin{aligned}\pi(\rho^{\mathfrak{M}}(\exists x.\, \varphi)) = \pi(\bigcup_{m \in M} \rho[m/x]^{\mathfrak{M}}(\varphi)) &\overset{1}{=} \bigcup_{m \in M} \pi(\rho[m/x]^{\mathfrak{M}}(\varphi)) \\ &\overset{\text{IH}}{=} \bigcup_{m \in M} (\pi \circ \rho[m/x])^{\mathfrak{M}}(\varphi) \\ &\overset{2}{=} \bigcup_{m \in M} (\pi \circ \rho)[\pi(m)/x]^{\mathfrak{M}}(\varphi) \\ &\overset{3}{=} \bigcup_{m' \in M} (\pi \circ \rho)[m'/x]^{\mathfrak{M}}(\varphi) \overset{4}{=} (\pi \circ \rho)^{\mathfrak{M}}(\forall x.\, \varphi)\end{aligned}$$

  Equalities (1) and (2) are given by Lemma 5.3.1. IH can be used because we are proving this for every M-valuation, including $\rho[m/x]$. Equality (3) holds because π is surjective. Equality (4) is again by definition of pattern interpretation ($\pi \circ \rho$ is an M-valuation).

■

Lemma 5.3.2 yields a direct manner how to prove that closed $\emptyset$-patterns are predicates:

Theorem 5.3.1. Let ψ be a *closed* $\emptyset$-pattern. Then ψ is an $\emptyset$-predicate (predicate pattern).

Proof. We want to prove that for every $\emptyset$-model $\mathfrak{M}$ we have $\rho^{\mathfrak{M}}(\psi) = \emptyset$ or $\rho^{\mathfrak{M}}(\psi) = M$. Let $\mathfrak{M}$ be an $\emptyset$-model. Because ψ is closed, we can fix some M-valuation ρ. By Lemma 5.3.2 we have

$$\pi(\rho^{\mathfrak{M}}(\psi)) = (\pi \circ \rho)^{\mathfrak{M}}(\psi)$$

for every permutation $\pi : M \to M$. Because ψ is closed, further $\pi(\rho^{\mathfrak{M}}(\psi)) = (\pi \circ \rho)^{\mathfrak{M}}(\psi) = \rho^{\mathfrak{M}}(\psi)$ (Proposition 2.2.1).

Let $A := \rho^{\mathfrak{M}}(\psi)$. This means that $\pi(A) = A$ for every permutation $\pi : M \to M$. Consider that this is only possible if $A = \emptyset$ or $A = M$. Assume for a contradiction that $\emptyset \subset A \subset M$. Choose $a \in A, b \notin A$ and consider the permutation $\pi(a) := b$, $\pi(b) := a$. But then obviously $\pi(A) \neq A$, which is a contradiction. ■

Now because we can consider only finite theories, by completeness of $\mathcal{H}$ w.r.t. predicates (Theorem 5.1.3) we have the following.

Theorem 5.3.2. Every $\emptyset$-theory Γ is $\mathcal{H}$-complete, i.e., $\mathcal{H}$ is complete w.r.t. the fragment of ML without symbols.

Proof. By Theorem 5.2.1 it suffices to prove that every finite $\emptyset$-theory Γ is $\mathcal{H}$-complete.

Let Γ be a finite $\emptyset$-theory. Γ can obviously be expressed as a conjuction $\gamma \equiv \bigwedge \Gamma$. By our if-and-only-if condition for $\mathcal{H}$-completeness we just need to prove for every $\varphi \in \textsc{Pattern}_{\emptyset}$ that $\{\gamma\}_{=} \vdash \varphi$ implies $\{\gamma\} \vdash \varphi$.

Let $\{\gamma\}_{=} \vdash \varphi$ for some $\varphi \in \textsc{Pattern}_{\emptyset}$. This is iff $\{\forall\gamma\}_{=} \vdash \forall\varphi$ by Proposition 4.2.1. Notice that $\forall\gamma$ is a closed $\emptyset$-pattern and thus $\forall\gamma$ is an $\emptyset$-predicate by Theorem 5.3.2. But $\mathcal{H}$ is complete w.r.t. predicate patterns (Theorem 5.1.3), i.e., $\{\forall\gamma\}_{=} \vdash \forall\varphi$ implies $\{\forall\gamma\} \vdash \forall\varphi$. Thus $\{\forall\gamma\} \vdash \forall\varphi$. Again by Proposition 4.2.1 this implies $\{\gamma\} \vdash \varphi$. ■

Thus any Σ-theory Γ with $\Sigma = \emptyset$ is $\mathcal{H}$-complete. This is important because now when we talk about completeness of $\mathcal{H}$, we can always assume w.l.o.g. that $|\Sigma| > 0$. This is more similar to the situation in FOL, where we always have at least one predicate symbol P (or "=" in the case of FOL with equality). As a concluding remark, Theorem 5.3.2 also answers the hanging question about Γ-deducts when Γ is an $\emptyset$-theory:

Proposition 5.3.1. Let Γ be an $\emptyset$-theory. Then every Γ-deduct is a Γ-predicate.

Proof. This is immediate from Theorem 5.3.2 because Γ-deducts are by definition closed $\emptyset$-patterns. ■

5.4 CONSISTENCY, SATISFIABILITY, AND COMPACTNESS

Consistency depends on the chosen proof system, not the chosen logic.

In this section we study consistency of ML theories. For any sufficiently strong proof system of matching logic such as $\mathcal{H}$, we can have the same definition of consistency as FOL, which has mostly the same properties. An $\mathcal{H}$-consistent theory means that you cannot derive the contradiction $\bot$ from this theory in $\mathcal{H}$. For example, the empty theory $\emptyset$ is consistent because $\mathcal{H}$ is sound (Theorem 4.2.1).

Definition 5.4.1 (Consistent theory). Let Γ be a Σ-theory. Then Γ is called *$\mathcal{H}$-consistent* (or simply consistent) iff $\Gamma \nvdash \bot$. ■

We often drop $\mathcal{H}$ in $\mathcal{H}$-consistency throughout the thesis. An inconsistent Σ-theory can prove every Σ-pattern (*explosion principle*), which is a consequence of what we have seen in Example 4.2.1. Therefore we can easily show that the notion of (in)consistency has the same equivalent definitions as most sound and complete systems for FOL do.

Lemma 5.4.1. Let Γ be a Σ-theory. The following three are equivalent.

(1) $\Gamma \vdash \bot$

(2) For every $\varphi \in \textsc{Pattern}_\Sigma$ it holds that $\Gamma \vdash \varphi$.

(3) There exists a closed pattern $\varphi \in \textsc{Pattern}_\Sigma$ such that $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$.

Proof. We show all the implications.

- (1⇒2) Let $\Gamma \vdash \bot$. The proof of $\Gamma \vdash \varphi$ is as follows. Apply proof of $\Gamma \vdash \bot$, notice that $\vdash \bot \to \varphi$ is an instance of (PT)[2], apply (MP) on $\bot$ and $\bot \to \varphi$.
- (2⇒3) Let $\Gamma \vdash \varphi$ for any pattern φ. Choose $\varphi_1 \equiv \exists x.\, x$ and $\varphi_2 \equiv \neg(\exists x.\, x)$. Thus $\Gamma \vdash \exists x.\, x$ and $\Gamma \vdash \neg(\exists x.\, x)$ by assumption.
- (3⇒1) Let $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$ for some φ. The proof of $\Gamma \vdash \bot$ is as follows. Apply proofs of $\Gamma \vdash \varphi$ and $\Gamma \vdash \neg\varphi$. Apply $\vdash \varphi \to (\neg\varphi \to \bot)$, which is an instance of (PT). Apply (MP) twice, first on φ and $\varphi \to (\neg\varphi \to \bot)$, then on $\neg\varphi$ and $(\neg\varphi \to \bot)$.

■

How can we be certain that this is an "appropriate" definition of consistency for ML? We should naturally aim for such a definition of consistency that unsatisfiable theories are inconsistent. Recall that we have seen in Example 2.4.1 that the unsatisfiable pattern $\neg x$ does not evaluate to $\emptyset$ (as $\bot$ does). This is fine because $\{\neg x\} \vdash \forall x.\, \neg x$ by (GEN)

2 This is the reason why we defined $\top$ as a propositional tautology, even though one can find a shorter definition such as $\exists x.\, x$.

and we have $\vdash \exists x.\, x$ by (Ex), which is the same as $\vdash \neg(\forall x.\, \neg x)$. Moreover, for theories with (Definedness) we can show that consistency exactly corresponds with satisfiability.

Theorem 5.4.1. Let Γ be a Σ-theory. Then $\Gamma_=$ is satisfiable iff $\Gamma_=$ is consistent.

Proof. We prove both implications.

($\Rightarrow$) Let $\mathfrak{M}$ be a model of Γ. Assume for a contradiction that $\Gamma_=$ is inconsistent, i.e., $\Gamma_= \vdash \bot$. Then by soundness of $\mathcal{H}$ we have $\Gamma_= \models \bot$. By definition of $\models$ we have $\mathfrak{M} \models \bot$, which means $\rho^{\mathfrak{M}}(\bot) = M$ for all M-valuations ρ, contradiction with $\rho^{\mathfrak{M}}(\bot) = \emptyset$.

($\Leftarrow$) Let $\Gamma_= \nvdash \bot$. By completeness of $\mathcal{H}$ w.r.t. $\Gamma_=$ we have $\Gamma_= \not\models \bot$, which yields there exists a model $\mathfrak{M}$ of $\Gamma_=$ such that $\mathfrak{M} \not\models \bot$, i.e., $\Gamma_=$ is satisfiable. ■

Observe that Theorem 5.4.1 and our extension by definedness lemma (Lemma 5.1.1) give a direct manner to show the well-known compactness property for ML! Compactness property for ML was not explicitly proved yet. What is more, here we show that it holds for any theory, even if it does not contain (Definedness).

Theorem 5.4.2 (Compactness property). Let Γ be any Σ-theory. Then Γ is satisfiable iff every finite subset $\Gamma_{\text{fin}} \subseteq \Gamma$ of Γ is satisfiable.

Proof. Lemma 5.1.1 yields that Γ is satisfiable iff $\Gamma_=$ is satisfiable. Theorem 5.4.1 yields $\Gamma_=$ is satisfiable iff $\Gamma_= \nvdash \bot$. But then by definition of $\vdash$ we have $\Gamma_= \nvdash \bot$ iff $\Gamma' \nvdash \bot$ for every finite subset $\Gamma' \subseteq \Gamma_=$. This is again by Theorem 5.4.1 iff every finite subset $\Gamma' \subseteq \Gamma_=$ is satisfiable iff every finite subset $\Gamma'' \subseteq \Gamma$ is satisfiable. ■

It should not be surprising the compactness property holds for ML after what we have seen in Section 3.1. However, it is still nice to have a direct and clear proof for our intuition. Does this mean that every $\mathcal{H}$-consistent theory has a model also for theories without (Definedness)? Unfortunately, for these theories we still only have the other implication that is proved by soundness of $\mathcal{H}$.

Proposition 5.4.1. Let Γ be a Σ-theory. If Γ is satisfiable, then Γ is consistent. ■

The other implication is usually proved by defining a canonical model for the so-called negation-complete theories, which are problematic in matching logic (see Section 5.5). It is not at all intuitive consistency implies satisfiability in ML, or equivalently, that unsatisfiablity implies inconsistency. One reason is that unsatisfiable patterns do not have to evaluate to $\emptyset$ (Example 2.4.1). In fact, even *closed* unsatisfiable patterns are not restricted to a few semantically equivalent edge cases. Unsatisfiable patterns can have complicated semantics. The following theorem is a new result confirming that closed unsatisfiable

patterns can evaluate to any strict subset of a model $\mathfrak{M}$, not just $\emptyset$, or $M \setminus \{m\}$ for some $m \in M$, etc. The difficulty is that ML symbols can be *self-referencing*, i.e., they can call themselves.

Theorem 5.4.3. Let Σ be any signature such that $\sigma \in \Sigma_1$. There exists a closed unsatisfiable pattern $\xi \in \text{PATTERN}_\Sigma$ such that for every Σ-model $\mathfrak{M}$ and set $X \subset M$, there exists a model $\mathfrak{M}'$ with $\rho^{\mathfrak{M}'}(\xi) = X$ for every M-valuation ρ.

ξ can match all numbers in $\mathbb{R} \setminus \mathbb{Q}$ and still not be valid in any model!

Proof. Consider the pattern $\xi = \sigma(\neg\sigma(\top))$. First we prove that $\{\xi\}$ is unsatisfiable by showing that $\{\xi\}$ is inconsistent (Proposition 5.4.1). Here is the proof for $\{\xi\} \vdash \bot$:

1. $\sigma(\neg\sigma(\top))$	
2. $\neg\sigma(\top) \to \top$	(PT)
3. $\sigma(\neg\sigma(\top)) \to \sigma(\top)$	2. (FRAMING)
4. $\sigma(\top)$	1., 3. (MP)
5. $\neg\sigma(\neg\sigma(\top))$	4. (Lemma 4.2.4)
6. $\bot$	1., 5. (Lemma 5.4.1)

The rest is straightforward. Let $\mathfrak{M}$ be some Σ-model. Choose any $X \subset M$ and consider the model $\mathfrak{M}' = (M, I \cap \{\sigma^{\mathfrak{M}'}\})$ where we define $\sigma^{\mathfrak{M}'}(m) = X$ for all $m \in M$. Note that $M \setminus X \neq \emptyset$. For every M-valuation ρ we get

$$\rho^{\mathfrak{M}'}(\xi) = \sigma^{\mathfrak{M}'}(M \setminus \bigcup_{b \in M} \sigma^{\mathfrak{M}'}(b)) = \sigma^{\mathfrak{M}'}(M \setminus X) = X.$$

■

The consequence of Theorem 5.4.3 is, for example, that we cannot simply claim that $\models \psi \to \bot$ for a closed unsatisfiable pattern ψ. In proof-theoretical terms, this forbids the deduction property even if we restrict the conclusions to $\bot$ such as

$$\psi \vdash \bot \text{ implies } \vdash \psi \to \bot.$$

Simply choose $\psi \equiv \sigma(\neg\sigma(\top))$ (recall that $\{\sigma(\neg\sigma(\top))\} \vdash \bot$ but by soundness $\nvdash \sigma(\neg\sigma(\top)) \to \bot$). Notice that this is exactly the kind of reasoning we needed for Henkin's reduction (Theorem 5.1.1), where we used only this particular instance of the deduction property!

However, there is still some hope. Consistency implies satisfiability is equivalent with something that is much more intuitive, i.e., that adding a definition of a fresh symbol does not break consistency.

Theorem 5.4.4. The following two properties are equivalent:

(1) $\Gamma \nvdash \bot$ implies $\Gamma \not\models \bot$

(consistency implies satisfiability),

(2) $\Gamma \nvdash \bot$ implies $\Gamma_= \nvdash \bot$

(adding a definition does not break consistency).

Proof. We have $\Gamma \not\models \bot$ iff $\Gamma_= \not\models \bot$ (Lemma 5.1.1) and $\Gamma_= \not\models \bot$ iff $\Gamma_= \nvdash \bot$ (Proposition 4.2.2). ■

Thus we leave consistency implies satisfiability as a conjecture.

Conjecture 5.4.1. $\Gamma \nvdash \bot$ implies Γ is satisfiable ($\Gamma \not\models \bot$).

Note that consistency implies satisfibility alone might not be enough for proving that $\mathcal{H}$ is complete (unlike in FOL and FOL proof systems). Consistency implies satisfiability overlaps with completeness only if every predicate is a deduct, as the following theorem shows. The proof of the theorem uses a trick that we can focus on Γ-predicates when dealing with completeness, together with the conventional trick from Henkin's reduction (Theorem 5.1.1):

Theorem 5.4.5 (Henkin's reduction in ML). Let us assume for any theory Γ that every closed Γ-predicate is a Γ-deduct. Then the following two statements are equivalent:

(1) $\mathcal{H}$ is complete.

(2) Every $\mathcal{H}$-consistent theory is satisfiable.

Proof. ($\Rightarrow$) Let $\Gamma \nvdash \bot$. Because $\mathcal{H}$ is complete, we automatically get $\Gamma \not\models \bot$. By definition of $\models$ there is a model $\mathfrak{M}$ of Γ such that $\mathfrak{M} \not\models \bot$, i.e., Γ is satisfiable.

($\Leftarrow$) Notice that it is enough to prove completeness of $\mathcal{H}$ for closed Γ-*predicates*, i.e., the following two are trivially equivalent:

(a) $\mathcal{H}$ is complete,

(b) For every Σ-theory Γ, every Γ-predicate φ, $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$.

This is because if $\Gamma \models \varphi$, trivially φ is a Γ-predicate (φ is valid in Γ). Let us prove (1) by proving the contraposition of (b), i.e., let Γ be a Σ-theory, φ a Γ-predicate and assume $\Gamma \nvdash \varphi$. First we show that $\Gamma \cup \{\neg\varphi\}$ is satisfiable. Assume $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable. Therefore by the contraposition of (2) we get $\Gamma \cup \{\neg\varphi\} \vdash \bot$. Because φ is a Γ-predicate, also $\neg\varphi$ is a Γ-predicate by Proposition 2.4.2. But then our extra assumption says that $\neg\varphi$ is a Γ-deduct. Thus by definition of Γ-deducts we get $\Gamma \vdash \neg\varphi \to \bot$. By (PT) we have $\Gamma \vdash (\neg\varphi \to \bot) \to \varphi$, thus obviously $\Gamma \vdash \varphi$, contradiction. Thus $\Gamma \cup \{\neg\varphi\}$ is satisfiable, i.e., there is some model $\mathfrak{M} \models \Gamma \cup \{\neg\varphi\}$. This means that $\mathfrak{M} \models \Gamma$ and $\mathfrak{M} \not\models \varphi$, i.e., $\Gamma \not\models \neg\varphi$. ■

The trouble with Henkin's reduction is even if we *had it*, we would probably need to deal with *negation-complete* theories. Negation-complete theories are problematic in ML. This is discussed in the next section.

5.5 NEGATION-COMPLETE THEORIES

One of the widely-known techniques in completeness proofs is finding how to construct a *canonical model* for any given consistent theory. In FOL, this is done by extending the given (consistent) theory to a (witnessed) *negation-complete theory* that can derive any closed formula or its negation [13, p. 78]:

Definition 5.5.1 (Negation-complete Σ-theory). A Σ-theory Γ is called *negation-complete* if Γ is consistent and for any closed pattern ψ holds either $\Gamma \vdash \psi$ or $\Gamma \vdash \neg\psi$. ■

In fact, maximally consistent FOL theories are negation-complete FOL theories, which can be easily proved using the deduction property. Since ML is a variant of FOL (Section 3.1), it is a natural question to ask whether such a class of theories actually exists in ML. There are indeed theories that can be extended to negation-complete ML theories:

Theorem 5.5.1. Let Ξ^+ be a maximally consistent set in some signature Σ and containing the axioms $\{\forall x.\, x, \lceil x \rceil\}$. Then Ξ^+ is negation-complete.

Proof. First we show some Ξ^+ exists. The theory $\Xi = \{\forall x.\, x, \lceil x \rceil\}$ is consistent because it is satisfiable (Proposition 5.4.1). For example, consider $\mathfrak{M} : M = \{0\}, \lceil\cdot\rceil^{\mathfrak{M}}(m) = M$ for all $m \in M$, clearly we have $\mathfrak{M} \models \Xi$. Now extend Ξ to some maximally $\mathcal{H}$-consistent set Ξ^+ in the conventional manner; the set Ξ^+ must exist for the same reasons as in FOL (System $\mathcal{H}$ is a Hilbert-style proof system).

Let us show that for every closed pattern $\varphi \in \text{PATTERN}_\Sigma$, either $\Xi^+ \vdash \varphi$ or $\Xi^+ \vdash \neg\varphi$. Assume for a contradiction that both $\Xi^+ \nvdash \varphi$ and $\Xi^+ \nvdash \neg\varphi$. By maximality this means $\Xi^+ \cup \{\varphi\} \vdash \bot$. Consider that φ is a Ξ^+-predicate because $\forall x.\, x \in \Xi^+$, i.e., for every $\mathfrak{M} \models \Xi^+$ we have $|M| = 1$. Because $\lceil x \rceil \in \Xi^+$, by Theorem 4.2.4 the pattern ψ is a Ξ^+-deduct. From this follows $\Xi^+ \vdash \psi \to \bot$, i.e., $\Xi^+ \vdash \neg\psi$ because $\vdash (\psi \to \bot) \to \neg\psi$ by (PT), contradiction.

■

However, there is a very simple example showing that not all ML theories can be extended to a negation-complete theory, even if we consider theory extensions instead of supersets (Definition 5.1.2):

Example 5.5.1. Consider the Σ-theory $\Gamma = \{\lceil x \rceil, \lceil \lambda \rceil, \lceil \neg\lambda \rceil\}$ where $\Sigma = \{\lambda\}$. Clearly Γ is satisfiable; consider

$$\mathfrak{M} : M = \{0,1\}, \lambda^{\mathfrak{M}} = \{0\}, \lceil\cdot\rceil^{\mathfrak{M}}(m) = M \text{ for every } m \in M,$$

which is indeed a model of Γ. Moreover, Γ is consistent (Theorem 5.4.1). Assume for a contradiction there is some negation-complete extension Γ^+ of Γ. By definition Γ^+ is consistent and because $\Gamma^+ \vdash \lceil x \rceil$, there is also some model $\mathfrak{M}^+$ of Γ^+ (Theorem 5.4.1). By negation-completeness

of Γ^+ we get that $\Gamma^+ \vdash \lambda$ or $\Gamma^+ \vdash \neg\lambda$ because λ is closed. Thus soundness of $\mathcal{H}$ yields either $\mathfrak{M}^+ \models \lambda$ or $\mathfrak{M}^+ \models \neg\lambda$, which is a contradiction because we have both $\mathfrak{M}^+ \models \lceil\lambda\rceil$ and $\mathfrak{M}^+ \models \lceil\neg\lambda\rceil$ ($\emptyset \subset \rho^{\mathfrak{M}^+}(\lambda) \subset M^+$ for every M^+-valuation ρ). ■

Notice that Example 5.5.1 applies to any sound proof system for ML. In fact, this shows a remarkable difference from FOL that theories of models $\mathsf{Th}(\mathfrak{M})$ are generally *not* negation-complete for any sound proof system of matching logic.

Definition 5.5.2 (Theory of a model $\mathfrak{M}$). Let $\mathfrak{M}$ be a Σ-model. Then

$$\mathsf{Th}(\mathfrak{M}) = \{\varphi \in \textsc{Pattern}_\Sigma \mid \mathfrak{M} \models \varphi\}$$

is called the theory of $\mathfrak{M}$. ■

Theories of models are obviously satisfiable because $\mathfrak{M} \models \mathsf{Th}(\mathfrak{M})$ by definition. Thus they are also consistent by Proposition 5.4.1. However, theories of models are not negation-complete. If you take the model $\mathfrak{M}$ from Example 5.5.1, then we have both $\mathsf{Th}(\mathfrak{M}) \not\vdash \neg\lambda$ and $\mathsf{Th}(\mathfrak{M}) \not\vdash \lambda$. How can this be? The explanation is that λ is not an $\mathfrak{M}$-predicate. That is why neither λ nor $\neg\lambda$ is valid in $\mathfrak{M}$, which excludes both from $\mathsf{Th}(\mathfrak{M})$.

We would like to find an alternative definition for negation-complete theories with similar properties as negation-complete theories in FOL, guaranteeing that

(C1) every theory can be extended to this kind of a negation-complete theory, and

(C2) theories of models would be negation-complete in this new sense.

To achieve this, it is obvious that we have to have the deduction property and patterns that are either true or false. Since we conjecture that Γ-deducts are exactly Γ-predicates (and vice versa), the following definition should fix both of our goals.

Definition 5.5.3 (Weakly negation-complete theory). Let Γ be a Σ-theory. If Γ is consistent and for every Γ-deduct φ we have $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$, then Γ is called *weakly negation-complete*. ■

Indeed, our definition of weakly negation-complete theories satisfies the condition (C1) because we can use the deduction property:

Theorem 5.5.2. Maximally consistent theories are weakly negation-complete.

Proof. Let Γ^+ be a maximally consistent theory. Assume for a contradiction that Γ^+ is not weakly negation-complete. Thus there is some Γ^+-deduct ψ such that $\Gamma^+ \not\vdash \psi$ and $\Gamma^+ \not\vdash \neg\psi$. Maximality implies that $\Gamma^+ \cup \{\psi\} \vdash \bot$. But then by definition of Γ^+-deducts $\Gamma^+ \vdash \psi \to \bot$. Consider $\vdash (\psi \to \bot) \to \neg\psi$ by (PT). Thus $\Gamma^+ \vdash \neg\psi$. Contradiction. ■

Corollary 5.5.1. Every consistent theory Γ can be extended to a weakly negation-complete theory. ■

As for the condition (C2), the situation is a little more complicated. Even though it is trivial to see that $\mathsf{Th}(\mathfrak{M})$ is an $\mathcal{H}$-complete theory, we do not know for every $\mathsf{Th}(\mathfrak{M})$ whether $\mathsf{Th}(\mathfrak{M})$-deducts are $\mathsf{Th}(\mathfrak{M})$-predicates (Conjecture 5.1.1). We can prove the condition (C2) only for models that satisfy (DEFINEDNESS):

Theorem 5.5.3. Let $\mathfrak{M}$ be a Σ-model with $\mathfrak{M} \models \lceil x \rceil$. Then $\mathsf{Th}(\mathfrak{M})$ is weakly negation-complete.

Proof. Let us prove that for every $\mathsf{Th}(\mathfrak{M})$-deduct ψ, either $\mathsf{Th}(\mathfrak{M}) \vdash \psi$ or $\mathsf{Th}(\mathfrak{M}) \vdash \neg\psi$. We will show a stronger claim that for every $\mathsf{Th}(\mathfrak{M})$-deduct ψ, either $\psi \in \mathsf{Th}(\mathfrak{M})$ or $\psi \notin \mathsf{Th}(\mathfrak{M})$.

Because (DEFINEDNESS) $\in \mathsf{Th}(\mathfrak{M})$, we know that every $\mathsf{Th}(\mathfrak{M})$-deduct is a closed $\mathsf{Th}(\mathfrak{M})$-predicate by Theorem 4.2.4. Thus it suffices to show that for every closed $\mathsf{Th}(\mathfrak{M})$-predicate ψ we have $\psi \in \mathsf{Th}(\mathfrak{M})$ or $\neg\psi \in \mathsf{Th}(\mathfrak{M})$. This is obvious because ψ is an $\mathfrak{M}$-predicate by $\mathfrak{M} \models \mathsf{Th}(\mathfrak{M})$ and for every closed $\mathfrak{M}$-predicate we have $\mathfrak{M} \models \psi$ or $\mathfrak{M} \models \neg\psi$. ■

For models that do not satisfy (DEFINEDNESS) we leave the condition (C2) as a conjecture, which can be proved using the same idea as Theorem 5.5.3 if every Γ-deduct is a Γ-predicate (Conjecture 5.1.1):

Conjecture 5.5.1. $\mathsf{Th}(\mathfrak{M})$ is weakly negation-complete for every Σ-model $\mathfrak{M}$.

5.6 OPEN LEADS

We have seen in Theorem 5.4.5 that we can use Henkin's reduction for ML if we learn more about deducts. So far, this is what we know about them:

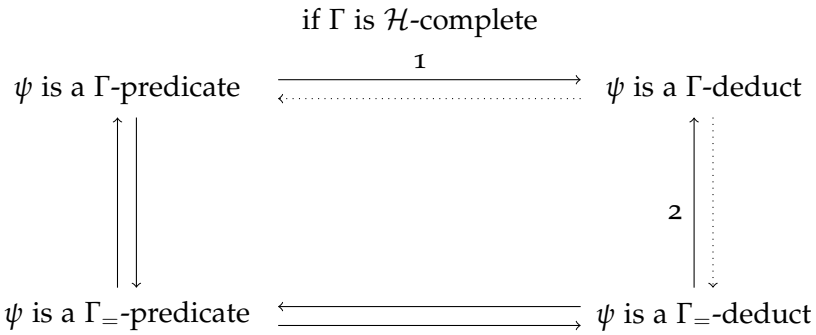


Figure 5.1: Our current knowledge about Γ-deducts assuming Γ is a Σ-theory and $\psi \in \text{PATTERN}_\Sigma$ is *closed*. (1) and (2) are equivalent arrows, which are proved to hold if Γ is $\mathcal{H}$-complete (Theorem 5.1.4). The dotted arrows are also equivalent and represent Conjecture 5.1.1.

We have shown that the arrow (1) holds only for $\mathcal{H}$-complete theories. If we proved the arrow (1) or (2) for every theory Γ, Theorem 5.4.5 says that completeness of $\mathcal{H}$ is equivalent to claiming that every $\mathcal{H}$-consistent theory has a model. This would greatly simplify things. Until then, it is unclear if this direction has more to offer.

There are two more directions which could lead to new results. First, recall that a theory Γ is $\mathcal{H}$-complete iff $\Gamma_=$ is a conservative extension of Γ (Theorem 5.1.2). Conservative extensions are tricky to work with directly but maybe there are some techniques that we have not tried. Second, we still have not looked at constructing some form of a *canonical model*.

We mentioned that we cannot copy the construction of canonical models from FOL because we do not have negation-complete theories. That is why we have to look for inspiration elsewhere, namely in modal logic. Modal logic has much in common with matching logic and offers two insights into completeness of $\mathcal{H}$:

(1) Some modal logics are known to be incomplete [2, 7]. If $\mathcal{H}$ is complete and ML can capture these logics, would it be a contradiction? This depends very much on what kind of capturing we would have and requires further research.

(2) Modal logic has a great deal of techniques for constructing canonical models that are different from the classical FOL construction. We might learn from these techniques how to construct a canonical model for ML.

The next chapter deals with how we can approach (2) and where it could potentially lead.

6

CANONICAL MODELS FOR EQUALITY EXTENSIONS

In Chapter 5 we looked at ways how to approach completeness of $\mathcal{H}$ using conservative extensions. Now we would like to turn to *canonical models*. Canonical models are in various forms a widely-known technique for proving completeness of proof systems (see e.g. [14] or [3]). Recall that in FOL, canonical models are built using negation-complete theories. In Section 5.5 we saw that these are problematic in ML. We have to look elsewhere for inspiration, namely into modal logic [5]. In this chapter we show a new technique how to construct canonical models for *equality extensions* that builds upon the theory developed in [10], which used canonical models from [5] to show local completeness of $\mathcal{H}$. Our contribution extends their construction as follows. For any given *consistent* equality extension $\Gamma_=$, we show a construction of a model $\mathfrak{M}_{\Gamma_=}$ such that

$$\Gamma_= \vdash \varphi \text{ iff } \mathfrak{M}_{\Gamma_=} \models \varphi.$$

In FOL this is sometimes called Henkin's theorem [13, p. 78].

Observe that if we could construct a canonical model $\mathfrak{M}_\Gamma$ for any consistent theory Γ, proving completeness of $\mathcal{H}$ would be straightforward. This is illustrated by the following example.

Example 6.0.1. We want to show $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$ by contraposition. Let $\Gamma \nvdash \varphi$, i.e., Γ is consistent. If we can construct the canonical model $\mathfrak{M}_\Gamma$, by construction $\mathfrak{M}_\Gamma \models \Gamma$ and also $\mathfrak{M}_\Gamma \not\models \varphi$, but then by definition of $\models$ we have $\Gamma \not\models \varphi$. ■

We will see how we can do this for equality extensions $\Gamma_=$. Our construction is thus an alternative proof of the fact that equality extensions are $\mathcal{H}$-complete. It also shows exactly what role (Definedness) semantically plays in the question of completeness, which could help trigger new developments in solving this problem. This is how we proceed:

(1) We make an overview of the existing theory from [10].

(2) We present a new technique how to use this theory to construct $\mathfrak{M}_{\Gamma_=}$ in the last Section 6.3.

6.1 LOCAL CONSISTENCY

Even if we had an analogoue of negation-complete theories (Section 5.5), it is not exactly clear how we could use them to construct a canonical model. The reason is that ML is not two-valued, which

is problematic. Suppose we have some theory Γ and pattern φ such that $\Gamma \nvdash \varphi$. This does not tell us much about φ even if $\mathcal{H}$ is complete! Notice that $\Gamma \not\models \varphi$ only means that φ does not match every element for some valuation. We do not know what exactly φ matches. The end of Section 5.4 hints in the proof of Theorem 5.4.5 that we can restrict ourselves to proving completeness for Γ-predicates. However, Γ-predicates are hard to work with because they depend on the theory Γ. For these reasons we are forced to find an inspiration for canonical models elsewhere, namely in *modal logic* [5]. The main idea is to switch from (general) provability $\vdash$ to the so-called *local provability*.

Definition 6.1.1 (Local provability [10]). Let Γ be a Σ-theory and $\varphi \in \textsc{Pattern}_\Sigma$. We say φ is *locally provable from* Γ, denoted $\Gamma \Vdash \varphi$, if there is some finite subset $\Gamma_{\text{fin}} \subseteq \Gamma$ such that $\vdash \bigwedge \Gamma_{\text{fin}} \to \varphi$. ■

There is a very intuitive correspondence between global and local deduction. Local deduction implies global deduction simply because we can repeatedly apply (MP).

Lemma 6.1.1 (Global and Local Deduction). If $\Gamma \Vdash \varphi$, then $\Gamma \vdash \varphi$.

Proof. Let $\Gamma \Vdash \varphi$. By definition there is some finite subset $\Gamma_{\text{fin}} \subseteq \Gamma$ such that $\vdash \bigwedge \Gamma_{\text{fin}} \to \varphi$. Notice that $\vdash ((\varphi_1 \wedge \varphi_2) \to \varphi) \leftrightarrow (\varphi_1 \to (\varphi_2 \to \varphi))$ by (PT). But then $\Gamma_{\text{fin}} \vdash \varphi$ by repeated application of (MP). Because $\Gamma_{\text{fin}} \subseteq \Gamma$, we get $\Gamma \vdash \varphi$. ■

Corollary 6.1.1. If $\Gamma \nvdash \bot$, then $\Gamma \nVdash \bot$. ■

Corollary 6.1.1 already suggests a very natural definition for local consistency, i.e., $\Gamma \nVdash \bot$. Instead of talking about maximally consistent sets, we shall talk about maximally *locally* consistent sets.

Definition 6.1.2 (Locally consistent set [10]). A set $\Gamma \subseteq \textsc{Pattern}_\Sigma$ is called *locally consistent* iff $\Gamma \nVdash \bot$. Furthermore, if for every $\varphi \in \textsc{Pattern}_\Sigma$ such that $\varphi \notin \Gamma$ we have $\Gamma \cup \{\varphi\}$ locally inconsistent, we say that Γ is *maximally locally consistent set* (MCS)[1]. ■

MCS's have several useful properties for building a canonical model. Most importantly, they are closed under negation, conjunction, and ML implication. This is possible because local provability is less strict than provability.

Lemma 6.1.2 (MCS properties [10]). Let Γ be an MCS in $(\Sigma, \textsc{Var})$. The following properties hold.

(1) $\varphi \in \Gamma$ iff $\Gamma \Vdash \varphi$;

(2) $\neg\varphi \in \Gamma$ iff $\varphi \notin \Gamma$;

(3) $\varphi_1 \wedge \varphi_2 \in \Gamma$ iff $\varphi_1 \in \Gamma$ and $\varphi_2 \in \Gamma$;

1 We use MCS instead of MLCS because we want to follow the naming in [10].

(4) $\varphi_1 \vee \varphi_1 \in \Gamma$ iff $\varphi_1 \in \Gamma$ or $\varphi_2 \in \Gamma$;

(5) $\varphi_1, \varphi_1 \to \varphi_2 \in \Gamma$ implies $\varphi_2 \in \Gamma$.

(6) $\forall x.\, \varphi \in \Gamma$ implies $\varphi[y/x] \in \Gamma$ for all $y \in \mathrm{VAR}$.

MCS properties allow the kind of reasoning we needed in Henkin's reduction (Theorem 5.1.1), especially when it comes to the deduction property:

Proposition 6.1.1. $\Gamma \Vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\} \Vdash \bot$.

Proof. ($\Rightarrow$) Let $\Gamma \Vdash \varphi$. Then there is some finite $\Gamma_{\mathsf{fin}} \subseteq \Gamma$ such that $\vdash \bigwedge \Gamma_{\mathsf{fin}} \to \varphi$. This is the same as $\vdash \bigwedge \Gamma_{\mathsf{fin}} \to (\neg\varphi \to \bot)$, by propositional reasoning $\vdash (\bigwedge \Gamma_{\mathsf{fin}}) \wedge \neg\varphi \to \bot$. However, $\Gamma_{\mathsf{fin}} \cup \{\neg\varphi\}$ is a finite subset of $\Gamma \cup \{\neg\varphi\}$, and thus $\Gamma \cup \{\neg\varphi\} \vdash \bot$ by definition of $\Vdash$.
($\Leftarrow$) This is symmetric to ($\Rightarrow$). ■

Similarly to Henkin's technique for constructing canonical models, we consider *witnessed sets* that add the so-called *witnesses*. If $\exists x.\, \varphi \in \Gamma$ for some MCS Γ, we do not know whether $\varphi[y/x] \in \Gamma$ for some $y \in \mathrm{VAR}$. This is different from property (6) in Lemma 6.1.2, which intuitively holds simply because of the property (5) and the rule (SUB) in $\mathcal{H}$. That is why we consider witnessed MCS:

Definition 6.1.3 (Witnessed MCS [10]). Let Γ be an MCS. We say that Γ is a *witnessed* MCS if for every $\exists x.\, \varphi \in \Gamma$ we have $(\exists x.\, \varphi) \to \varphi[y/x] \in \Gamma$ for some $y \in \mathrm{VAR}$. ■

The reason why we can focus on witnessed MCS's is that any MCS can be extended to a witnessed MCS (Lemma 6.1.3). Notice that in the following lemma we need to explicitly mention the variables we are using because we have to *extend them* for technical reasons.

Lemma 6.1.3 (Extension [10]). Let Γ be a locally consistent set in (Σ, VAR). Then there exists a witnessed MCS Γ^+ in (Σ, VAR^+) such that $\mathrm{VAR} \subseteq \mathrm{VAR}^+$ and $\Gamma \subseteq \Gamma^+$. ■

6.2 CANONICAL MODELS

Now as we finally know what witnessed MCS's are and what properties they have, we can define canonical models:

Definition 6.2.1 (Canonical model [10]). Let (Σ, VAR) be a signature. A canonical (Σ, VAR)-model $\mathfrak{K}$ is a Σ-model defined as

- $K = \{\Delta \mid \Delta \text{ is a witnessed MCS in } (\Sigma, \mathrm{VAR})\}$,
- $\Delta \in \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$ iff $\sigma(\varphi_1, \ldots, \varphi_n) \in \Delta$ for all $\varphi_i \in \Delta_i$ where $1 \leq i \leq n$.

■

Note that canonical models are well-defined because witnessed MCS's exist by Lemma 6.1.3. This is admittedly a very complex formalism. To show $\Delta \in \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$, we have to consider *all* patterns $\varphi_i \in \Delta_i$ whether $\sigma(\varphi_1, \ldots, \varphi_n) \in \Delta$. However, there is an easier and stronger description of how the symbol interpretations behave. Rather than considering all patterns in each Δ_i, the following lemma says that $\sigma(\varphi_1, \ldots, \varphi_n) \in \Delta$ already implies that corresponding Δ_i exists for each $1 \leq i \leq n$. What is more, φ_i will be contained in in Δ_i:

Lemma 6.2.1 (Existence [10]). Let Δ be a witnessed MCS and $\mathfrak{K}$ the canonical model in the corresponding signature. If $\sigma(\varphi_1, \ldots, \varphi_n) \in \Delta$, then there exist witnessed MCS's $\Delta_1, \ldots, \Delta_n \in K$ such that $\varphi_1 \in \Delta_1, \ldots, \varphi_n \in \Delta_n$ and $\Delta \in \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$. ■

Notice that the canonical model does not depend in any way on a theory Γ. That is where the so-called Γ-*generated model* steps in:

Definition 6.2.2 (Generated model [10]). Let Γ be a witnessed MCS in $(\Sigma, \textsc{Var})$ and $\mathfrak{K}$ be the canonical $(\Sigma, \textsc{Var})$-model. A Γ-generated model $\mathfrak{M}$ is a Σ-model such that M is the smallest set satisfying

- $\Gamma \in M$,
- if $\Delta \in M$ and there exist witnessed MCS's $\Delta_1, \ldots, \Delta_n \in K$ such that $\Delta \in \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$ for some $\sigma \in \Sigma_n$, then $\Delta_1, \ldots, \Delta_n \in M$,

and for every $\sigma \in \Sigma_n$ we have $\sigma^{\mathfrak{M}}(\Delta_1, \ldots, \Delta_n) := M \cap \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$. ■

This construction is again non-trivial, and this time we will give much more space for describing its properties. One can imagine the Γ-generated model as a countable "cut" through the canonical model (Figure 6.1). The cut is determined by Γ only: notice that Γ itself is a witnessed MCS, which serves as the basis of the (inductive) construction. Then we make "transitions" to other witnessed MCS simply by choosing those witnessed MCS's, which we formally need as arguments for our symbols.

The cut is indeed countable. To each set added to the generated model, there is a so-called generating path from the basis Γ to this set:

Definition 6.2.3 (Generating path). Let Γ be a witnessed MCS in $(\textsc{Var}, \Sigma)$, $\mathfrak{K}$ the canonical model in $(\textsc{Var}, \Sigma)$ and $\mathfrak{M}$ some Γ-generated model. A *generating path relation* $P : M \times \Sigma^*$ is defined inductively as follows:

- $P(\Gamma, \varepsilon)$.
- if $P(\Delta, \pi)$ and there exists $\Delta_1, \ldots, \Delta_n \in K$ such that

  $$\Delta \in \sigma^{\mathfrak{K}}(\Delta_1, \ldots, \Delta_n)$$

  for some $\sigma \in \Sigma_n$, then also $P(\Delta_i, \pi\sigma)$ for all $1 \leq i \leq n$.

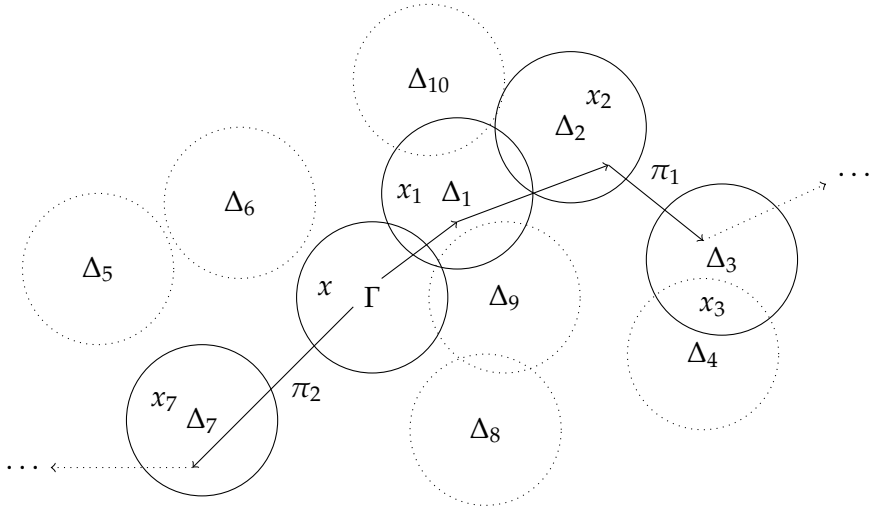


Figure 6.1: A simplified sketch of the carrier set in *some* Γ-generated model $\mathfrak{M}$. Each circle is a witnessed MCS in the canonical model $\mathfrak{K}$. Dashed circles are witnessed MCS from $\mathfrak{K}$ that are not included in $\mathfrak{M}$. Regular circles are witnessed MCS *included* in $\mathfrak{M}$ by definition of Γ-generated models, i.e., each one of these MCS is added by some generating path π and is represented by some unique variable (Lemma 6.2.3).

We say that π is a generating path of a witnessed MCS $\Delta \in M$ if $P(\Delta, \pi)$. ■

Notice that the relation P exactly copies the definition of a generated model (Definition 6.2.2). That is why obviously for every set $\Delta \in M$, there is at least one generating path π of Δ, i.e., $P(\Delta, \pi)$. Especially for Γ we have $P(\Gamma, \varepsilon)$ by construction. Given a generating path π, we define a *path context*. Path contexts are "dummy" symbol contexts Definition 4.2.1 such that the path to $\Box$ is exactly a generating path:

Definition 6.2.4 (Path context). A *path context* C_π for a generating path π is defined inductively as follows.

- $C_\varepsilon \equiv \Box$ is a path context,
- $C_{\pi\sigma} = C_\pi[\sigma(\underbrace{\top, \ldots, \Box, \ldots, \top}_{n})]$ for a path context C_π and $\sigma \in \Sigma_n$.

■

Path contexts are interesting because of the fact that they are contained in the basis Γ of the Γ-generated model. This means that representatives of *every* set we add to the generated model are already given at the start of the construction:

Lemma 6.2.2 ([10]). Let Γ be a witnessed MCS, let $\mathfrak{M}$ be the Γ-generated model and consider some $\Delta \in M$. If Δ has a generating path π, then $C_\pi[\varphi] \in \Gamma$ for any pattern $\varphi \in \Delta$. ■

This has several immediate consequences. For example, each witnessed MCS in the generated model contains unique variables. In other words, a single variable cannot be in two different MCS's:

Lemma 6.2.3 (Singleton variables [10]). Let Γ be a witnessed MCS in $(\Sigma, \textsc{Var})$ and consider the Γ-generated model $\mathfrak{M}$. For every $x \in \textsc{Var}$ and $\Delta_1, \Delta_2 \in M$ we have that $x \in \Delta_1 \cap \Delta_2$ implies $\Delta_1 = \Delta_2$. ■

This means that each set contained in a generated model is represented not only by a generating path, but also a unique variable:

Lemma 6.2.4. Let Γ be a witnessed MCS in $(\Sigma, \textsc{Var})$ and consider the Γ-generated model $\mathfrak{M}$. For every $\Delta \in M$ there is at least one variable $y \in \textsc{Var}$ such that $y \in \Delta$.

Proof. Every $\Delta \in M$ must contain a pattern α-equivalent to $\exists x.\, x$ because $\vdash \exists x.\, x$ by definition of $\mathcal{H}$ and Δ is an MCS. Since Δ is also witnessed we have $(\exists x.\, x) \to x[y/x] \in \Delta$ for some $y \in \textsc{Var}$, but then by MCS properties $y \in \Delta$. ■

Corollary 6.2.1 (Representatives). Let Γ be a witnessed MCS in $(\textsc{Var}, \Sigma)$ and $\mathfrak{M}$ be the Γ-generated model. Then for every $\Delta \in M$ there is at least one variable $y \in \textsc{Var}$ such that $y \in \Delta$ and y is *unique* in $\mathfrak{M}$. ■

What are Γ-generated models good for? We would like them to connect syntax with semantics. This concept is often called in modal logic as a *truth lemma* (see, e.g., [4, p. 201]). Ideally, we would like to find a generated model $\mathfrak{M}$ and an M-valuation ρ for the generated model such that the following holds: for every witnessed MCS $\Delta \in M$ and every $\varphi \in \textsc{Pattern}_\Sigma$ we have

$$\varphi \in \Delta \text{ iff } \Delta \in \rho^{\mathfrak{M}}(\varphi).$$

Is this possible? There is a fundamental problem right at the core. Obviously we need a valuation ρ such that $x \in \Delta$ iff $\rho(x) = \Delta$. Lemma 6.2.3 guarantees that such a valuation would be a function. However, we have no guarantee that this function would be total. In other words, we do not know whether each variable is *covered* by some witnessed MCS in the generated model:

Definition 6.2.5 ($\mathfrak{M}$-covered variable). Let Γ be a witnessed MCS and $\mathfrak{M}$ the Γ-generated model. Then the variable $x \in \textsc{Var}$ is called to be $\mathfrak{M}$-*covered* (or simply covered) iff $x \in \Delta$ for some witnessed MCS $\Delta \in M$. ■

That is why we have to come up with yet another definition that "completes" the generated models by filling holes for those variables which possibly might not be covered. We call them *completed models*.

Definition 6.2.6 (Completed model). Let Γ be a witnessed MCS in $(\textsc{Var}, \Sigma)$ and $\mathfrak{M}$ be the Γ-generated model. Let $* \notin M$ be some fresh element. A Γ-*completed model* is any model $\mathfrak{M}^*$ such that:

- $M^* = \begin{cases} M \cup \{*\} & \text{some } x \in \textsc{Var} \text{ is not } \mathfrak{M}\text{-covered} \\ M & \text{otherwise} \end{cases}$
- $\sigma^{\mathfrak{M}^*}(m_1, \ldots, m_n) =$

$$= \begin{cases} \emptyset & m_i = * \text{ for some } 1 \leq i \leq n \\ \sigma^{\mathfrak{M}}(m_1, \ldots, m_n) \cup \{*\} & m_i = \Gamma \text{ for some } 1 \leq i \leq n, * \in M^* \\ \sigma^{\mathfrak{M}}(m_1, \ldots, m_n) & \text{otherwise} \end{cases}$$

■

Notice that if a Γ-generated model $\mathfrak{M}$ has all variables covered, then $\mathfrak{M}$ is a (unique) Γ-completed model. We only add $*$ if some variable is not $\mathfrak{M}$-covered, in which case there can be multiple completed models (depending on the chosen $*$). This will be very important in our construction. Once we have completed models, we can easily define the so-called *completed valuation*, for which we already gave the intuition:

Definition 6.2.7 (Completed valuation). Let Γ be a witnessed MCS and $\mathfrak{M}^*$ a Γ-completed model. Then the *completed valuation* of $\mathfrak{M}^*$ is an M^*-valuation ρ defined as:

$$\rho(x) = \begin{cases} \Delta & x \in \Delta \\ * & \text{otherwise} \end{cases}$$

■

Completed valuations are well-defined because

- by Lemma 6.2.3 if $x \in \Delta_1$ and $x \in \Delta_2$, then $\Delta_1 = \Delta_2$.
- $\rho(x) = *$ iff x is not covered, in which case $*$ will be in a completed model.

Γ-completed models connect syntax and semantics, i.e., the so-called truth lemma holds for Γ-completed models and completed valuations:

Lemma 6.2.5 (Truth Lemma [10]). Let Γ be a witnessed MCS in $(\textsc{Var}, \Sigma)$. Consider a Γ-completed model $\mathfrak{M}$ and the corresponding completed valuation ρ.

For every witnessed MCS $\Delta \in M$, every $\varphi \in \textsc{Pattern}_\Sigma$ we have

$$\varphi \in \Delta \text{ iff } \Delta \in \rho^{\mathfrak{M}}(\varphi).$$

■

This lemma is the main idea behind proving local completeness of $\mathcal{H}$, which we discussed in Section 4.2.4. We do not present the proof here, interested readers are referred to [10] for details.

6.3 NEW RESULTS

We want to use the theory developed in this chapter for constructing a canonical model *of* a given consistent theory Γ. In this section we show how to do it for equality extensions. There is only one crucial observation to be made. For simplicity assume that some witnessed MCS Γ contains (DEFINEDNESS) (then the same argument holds for an equality extension). Then we can show that the Γ-generated model will have all variables covered! That is, the Γ-generated model will also be a Γ-completed model:

Lemma 6.3.1 (Variable Covering). Let Γ be a witnessed MCS in (VAR, Σ) and consider the Γ-generated model $\mathfrak{M}$. If $\forall x.\lceil x\rceil \in \Gamma$, then for every $x \in \mathrm{VAR}$ we have $x \in \Delta$ for some $\Delta \in M$ (x is $\mathfrak{M}$-covered).

Proof. Let $y \in \mathrm{VAR}$ be any variable. We have $\vdash (\forall x.\lceil x\rceil) \to \lceil x\rceil[y/x]$ by (SUB). By MCS properties $\lceil y\rceil \in \Gamma$.

But then by the Existence Lemma (Lemma 6.2.1) there exists some $\Delta_1 \in K$ such that $y \in \Delta_1$ and $\Gamma \in \lceil\cdot\rceil^{\mathfrak{K}}(\Delta_1)$. Since $\Gamma \in M$ and there exists $\Delta_1 \in K$ such that $\Gamma \in \lceil\cdot\rceil^{\mathfrak{K}}(\Delta_1)$, by construction of the Γ-generated model we also have $\Delta_1 \in M$. This means $y \in \Delta_1$ for some $\Delta_1 \in M$, i.e., y is $\mathfrak{M}$-covered. ■

Corollary 6.3.1. Let Γ be a witnessed MCS. If $\forall x.\lceil x\rceil \in \Gamma$, then a Γ-generated model is also a Γ-completed model. ■

Most importantly, Corollary 6.3.1 says that a Γ-completed valuation ρ always returns a witnessed MCS (and never $*$). But now we can easily prove for theories Γ containing the axiom (DEFINEDNESS) that there is a generated model $\mathfrak{M}$ satisfying Henkin's theorem:

Theorem 6.3.1. Let Γ be a consistent Σ-theory containing $\forall x.\lceil x\rceil$. Then there exists a witnessed MCS $\Gamma^+ \supseteq \Gamma$ such that a Γ^+-completed model $\mathfrak{M}$ satisfies:

$$\Gamma \vdash \varphi \text{ iff } \mathfrak{M} \models \varphi.$$

Proof. Let Γ be a (globally) consistent Σ-theory. By Lemma 4.2.4 we have $\Gamma \vdash \neg C[\neg\psi]$ for any nested symbol context C in the signature of Γ and ψ such that $\Gamma \vdash \psi$. But then

$$\Gamma' = \Gamma \cup \{\neg C[\neg\psi] \in \mathrm{PATTERN}_\Sigma \mid \Gamma \vdash \psi\}$$

is (globally) consistent because it is obviously a conservative extension of Γ. Also $\Gamma'' = \Gamma' \cup \{\neg\varphi \mid \Gamma \nvdash \varphi\}$ is locally consistent by Proposition 6.1.1.

Extend Γ'' to a witnessed MCS Γ^+ and consider any Γ^+-completed model $\mathfrak{M}$. We know that $\forall x.\lceil x\rceil \in \Gamma \subseteq \Gamma^+$, but then by the covering lemma (Lemma 6.3.1), $\mathfrak{M}$ is also a generated model, i.e., $* \notin M$. Now take the completed M-valuation ρ. We will show that $\Gamma \vdash \psi$ implies

$\mathfrak{M} \models \psi$ and $\Gamma \nvdash \varphi$ implies $\mathfrak{M} \not\models \varphi$. We can assume w.l.o.g.that ψ and φ are closed (Lemma 2.4.1).

(1) Let $\Gamma \vdash \psi$, we can show $\mathfrak{M} \models \psi$. Because ψ is closed, we just need to show that $\Delta \in \rho^{\mathfrak{M}}(\psi)$ for every $\Delta \in M$. By Truth Lemma (Lemma 6.2.5) $\Delta \in \rho^{\mathfrak{M}}(\psi)$ iff $\psi \in \Delta$.

Assume for a contradiction that there is some $\Delta \in M$ such that $\psi \notin \Delta$. By MCS properties thus $\neg\psi \in \Delta$. Recall that Δ has at least one generating path π and by Lemma 6.2.2 $C_\pi[\neg\psi] \in \Gamma^+$. However, by construction we have $\neg C_\pi[\neg\psi] \in \Gamma^+$ because $C_\pi[\neg\psi] \in \textsc{Pattern}_\Sigma$. By MCS properties $C_\pi[\neg\psi] \notin \Gamma^+$, contradiction.

(2) Let $\Gamma \nvdash \varphi$. Then we can show $\mathfrak{M} \not\models \varphi$, e.g., by showing $\Gamma^+ \notin \rho^{\mathfrak{M}}(\varphi)$ (because this would mean $\rho^{\mathfrak{M}}(\varphi) \neq M$). By Truth Lemma (Lemma 6.2.5) this is iff $\varphi \notin \Gamma^+$. By construction $\neg\varphi \in \Gamma'' \subseteq \Gamma^+$ and thus by MCS properties $\varphi \notin \Gamma^+$. Therefore $\rho^{\mathfrak{M}}(\varphi) \neq M$ and $\mathfrak{M} \not\models \varphi$ by definition.

■

Corollary 6.3.2. For every consistent equality extension $\Gamma_=$ there is a model $\mathfrak{M}_{\Gamma_=}$ such that $\Gamma_= \vdash \varphi$ iff $\mathfrak{M}_{\Gamma_=} \models \varphi$.

Proof. Use the same technique as presented in Theorem 6.3.1 replacing (Definedness) with the corresponding fresh symbol. ■

7 CONCLUSION

We set out to answer whether System $\mathcal{H}$ is complete as a proof system w.r.t. all ML theories. Even though we did not give a final answer, we gave a summary of results useful for answering this question, found a very natural characterization of completeness, and exploited it to find new classes of $\mathcal{H}$-complete theories. Along the way, we also proved related results, such as the compactness property. We have also extended the existing theory with new concepts such as equality extensions, Γ-deducts, weakly negation-complete theories, or $\mathcal{H}$-consistency and showed that they mostly behave as expected. Regardless of the final answer to completeness of $\mathcal{H}$, the presented results will remain relevant and should (hopefully) bring some new light to this debate.

WHAT DID NOT WORK. We have seen that ML is an FOL variant in Chapter 3; in particular, these two logics have the same level of expressiveness. There are many connections, and some of them are useful for results in ML. At the end of Section 5.4, we saw that the two-valued Γ-predicates (resembling FOL semantics) are actually the only patterns "relevant" for completeness of $\mathcal{H}$. However, it currently seems that investigating completeness of $\mathcal{H}$ from an FOL perspective has reached a dead end. Unless we learn more about the counterparts of Γ-predicates, i.e., Γ-deducts, it is hard to imagine how to make FOL techniques work for completeness of $\mathcal{H}$. For example, in Section 5.5 we presented an analogue of negation-complete theories that had seemed promising; in the end we could not find out much more, even under the assumption that $\mathcal{H}$ is complete.

WHAT WORKED. We looked away from FOL and tried to prove many things directly in ML. First, the intuition that (DEFINEDNESS) "makes" $\mathcal{H}$ complete led to a more tractable if-and-only-if condition for completeness of $\mathcal{H}$: System $\mathcal{H}$ is complete iff every equality extension is a conservative extension. To the best of our knowledge, we have not seen such an approach in completeness proofs. What is more, this approach allowed us to focus on finite theories and prove completeness on a case-by-case basis. Second, the modal logic idea for canonical models led to a constructive proof that $\mathcal{H}$ is complete w.r.t. equality extensions. We now know what canonical models for equality extensions look like, no matter what these theories with equality specify. This is a new result, which did not follow directly from the results presented in [10].

FUTURE WORK. Our characterization of completeness depends on conservative extensions. Conservative extensions are difficult to handle without a complete proof system. Unless we find alternative techniques to work with conservative extensions, it seems that looking for inspiration in modal logic is the most promising direction for this problem. We saw that we could take canonical models in modal logic and turn them into canonical models of ML equality extensions. It might be possible to take this construction even further. For example, the definition of completed models could be tweaked to allow stronger versions of the truth lemma. On the other hand, there are also hints suggesting *incompleteness* of $\mathcal{H}$. Some modal logics are known to be incomplete (see, e.g., [2] or [12]). If we can define them as matching logic theories, what would it lead to? This requires further research.

BIBLIOGRAPHY

[1] Oskar Becker. *Zur Logik der Modalitäten*. Max Niemeyer Verlag, 1930.

[2] Johan F. A. K. van Benthem. "Two simple incomplete modal logics." In: *Theoria* 44.1 (Feb. 11, 2008), pp. 25–37. ISSN: 00405825, 17552567. DOI: 10.1111/j.1755-2567.1978.tb00830.x. URL: https://onlinelibrary.wiley.com/doi/10.1111/j.1755-2567.1978.tb00830.x (visited on 05/10/2022).

[3] Patrick Blackburn, Johan F. A. K. van Benthem, and Frank Wolter, eds. *Handbook of Modal Logic*. 1st edition. Amsterdam Boston: Elsevier Science, Dec. 25, 2006. 1260 pp. ISBN: 978-0-444-51690-9.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge: Cambridge University Press, Sept. 30, 2002. 578 pp. ISBN: 978-0-521-52714-9.

[5] Patrick Blackburn and Miroslava Tzakova. "Hybrid completeness." In: *Logic Journal of the IGPL* 6.4 (July 1998), pp. 625–650. ISSN: 1367-0751. DOI: 10.1093/jigpal/6.4.625. URL: https://academic.oup.com/jigpal/article-pdf/6/4/625/1878635/060625.pdf.

[6] Denis Bogdanas and Grigore Roşu. "K-Java: A Complete Semantics of Java." In: *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. POPL '15: The 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. Mumbai India: ACM, Jan. 14, 2015, pp. 445–456. ISBN: 978-1-4503-3300-9. DOI: 10.1145/2676726.2676982. URL: https://dl.acm.org/doi/10.1145/2676726.2676982 (visited on 05/01/2022).

[7] George Boolos and Giovanni Sambin. "An incomplete system of modal logic." In: *Journal of Philosophical Logic* 14.4 (Nov. 1, 1985), pp. 351–358. ISSN: 1573-0433. DOI: 10.1007/BF00649480. URL: https://doi.org/10.1007/BF00649480 (visited on 05/10/2022).

[8] Xiaohong Chen, Dorel Lucanu, and Grigore Roşu. "Matching logic explained." In: *Journal of Logical and Algebraic Methods in Programming* 120 (2021), p. 100638. ISSN: 2352-2208. DOI: https://doi.org/10.1016/j.jlamp.2021.100638. URL: https://www.sciencedirect.com/science/article/pii/S2352220821000018.

[9] Xiaohong Chen and Grigore Roşu. "Applicative matching logic." In: (July 31, 2019). URL: https://www.ideals.illinois.edu/handle/2142/104616 (visited on 05/09/2022).

[10] Xiaohong Chen and Grigore Roşu. *Matching mu-Logic (Technical Report)*. University of Illinois at Urbana-Champaign, Tech. Rep. 2019. URL: http://hdl.handle.net/2142/102281 (visited on 09/24/2021).

[11] Xiaohong Chen and Grigore Roşu. "Matching mu-Logic." In: *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'19)*. ACM/IEEE, June 2019, pp. 1–13. DOI: https://doi.org/10.1109/LICS.2019.8785675.

[12] Max J. Cresswell and George E. Hughes. *A New Introduction to Modal Logic*. 1st edition. London ; New York: Routledge, Sept. 12, 1996. 432 pp. ISBN: 978-0-415-12600-7.

[13] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic, 2nd Edition*. 2nd edition. New York: Springer, June 10, 1994. 301 pp. ISBN: 978-0-387-94258-2.

[14] Herbert B. Enderton. *A Mathematical Introduction to Logic*. 2nd edition. San Diego: Academic Press, Jan. 5, 2001. 336 pp. ISBN: 978-0-12-238452-3.

[15] Jean-Yves Girard. *Proofs and types*. Cambridge tracts in theoretical computer science 7. Cambridge [England] ; New York: Cambridge University Press, 1989. 176 pp. ISBN: 978-0-521-37181-0.

[16] Kurt Gödel. "Die Vollständigkeit der Axiome des logischen Funktionenkalküls." In: *Monatshefte für Mathematik und Physik* 37.1 (Dec. 1930), pp. 349–360. ISSN: 0026-9255, 1436-5081. DOI: 10.1007/BF01696781. URL: http://link.springer.com/10.1007/BF01696781 (visited on 04/27/2022).

[17] David Harel, Dexter Kozen, and Jerzy Tiuryn. *Dynamic logic*. Foundations of computing series. Cambridge, Mass. London: MIT Press, 2000. 459 pp. ISBN: 978-0-262-52766-8 978-0-262-08289-1.

[18] John Harrison. *Handbook of Practical Logic and Automated Reasoning*. 1st edition. Cambridge ; New York: Cambridge University Press, Apr. 13, 2009. 702 pp. ISBN: 978-0-521-89957-4.

[19] Chris Hathhorn, Chucky Ellison, and Grigore Roşu. "Defining the undefinedness of C." In: *ACM SIGPLAN Notices* 50.6 (Aug. 7, 2015), pp. 336–345. ISSN: 0362-1340, 1558-1160. DOI: 10.1145/2813885.2737979. URL: https://dl.acm.org/doi/10.1145/2813885.2737979 (visited on 05/01/2022).

[20] Leon Henkin. "The completeness of the first-order functional calculus." In: *Journal of Symbolic Logic* 14.3 (Sept. 1949), pp. 159–166. ISSN: 0022-4812, 1943-5886. DOI: 10.2307/2267044. URL: https://www.cambridge.org/core/product/identifier/S0022481200105675/type/journal_article (visited on 03/17/2022).

[21] Dexter Kozen. "Results on the propositional mu-calculus." In: *Theoretical Computer Science* 27.3 (1983), pp. 333–354. ISSN: 03043975. DOI: 10.1016/0304-3975(82)90125-6. URL: https://linkinghub.elsevier.com/retrieve/pii/0304397582901256 (visited on 05/01/2022).

[22] Saul A. Kripke. "A completeness theorem in modal logic." In: *Journal of Symbolic Logic* 24.1 (Mar. 1959), pp. 1–14. ISSN: 0022-4812, 1943-5886. DOI: 10.2307/2964568. URL: https://www.cambridge.org/core/product/identifier/S0022481200125058/type/journal_article (visited on 05/09/2022).

[23] *Matching Logic*. Matching Logic. URL: http://www.matching-logic.org/ (visited on 05/09/2022).

[24] Daejun Park, Andrei Stefănescu, and Grigore Roşu. "KJS: a complete formal semantics of JavaScript." In: *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI '15: ACM SIGPLAN Conference on Programming Language Design and Implementation. Portland OR USA: ACM, June 3, 2015, pp. 346–356. ISBN: 978-1-4503-3468-6. DOI: 10.1145/2737924.2737991. URL: https://dl.acm.org/doi/10.1145/2737924.2737991 (visited on 05/01/2022).

[25] Amir Pnueli. "The temporal logic of programs." In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. 18th Annual Symposium on Foundations of Computer Science (sfcs 1977). Providence, RI, USA: IEEE, Sept. 1977, pp. 46–57. DOI: 10.1109/SFCS.1977.32. URL: http://ieeexplore.ieee.org/document/4567924/ (visited on 05/01/2022).

[26] Grigore Rosu, Andrei Stefanescu, Stefan Ciobâcá, and Brandon M. Moore. "One-Path Reachability Logic." In: *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*. 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. ISSN: 1043-6871. June 2013, pp. 358–367. DOI: 10.1109/LICS.2013.42.

[27] Grigore Roşu. "Matching logic." In: *Logical Methods in Computer Science* 13.4 (Dec. 2017), pp. 1–61. DOI: http://arxiv.org/abs/1705.06312.

[28] Grigore Roşu and Traian Florin Şerbănuţă. "K Overview and SIMPLE Case Study." In: *Electronic Notes in Theoretical Computer Science* 304 (June 2014), pp. 3–56. ISSN: 15710661. DOI: 10.1016/j.entcs.2014.05.002. URL: https://linkinghub.elsevier.com/retrieve/pii/S1571066114000383 (visited on 05/01/2022).

[29] Andrei Stefănescu, Daejun Park, Shijiao Yuwen, Yilong Li, and Grigore Roşu. "Semantics-based program verifiers for all languages." In: *ACM SIGPLAN Notices* 51.10 (Dec. 5, 2016), pp. 74–91. ISSN: 0362-1340, 1558-1160. DOI: 10.1145/3022671.2984027.

URL: https://dl.acm.org/doi/10.1145/3022671.2984027 (visited on 04/26/2022).

[30] Alfred Tarski. *The concept of truth in formalized languages*. New York, NY: Springer Berlin Heidelberg, 2016. ISBN: 978-3-319-32614-6.